

Microsoft SharePoint Monitoring



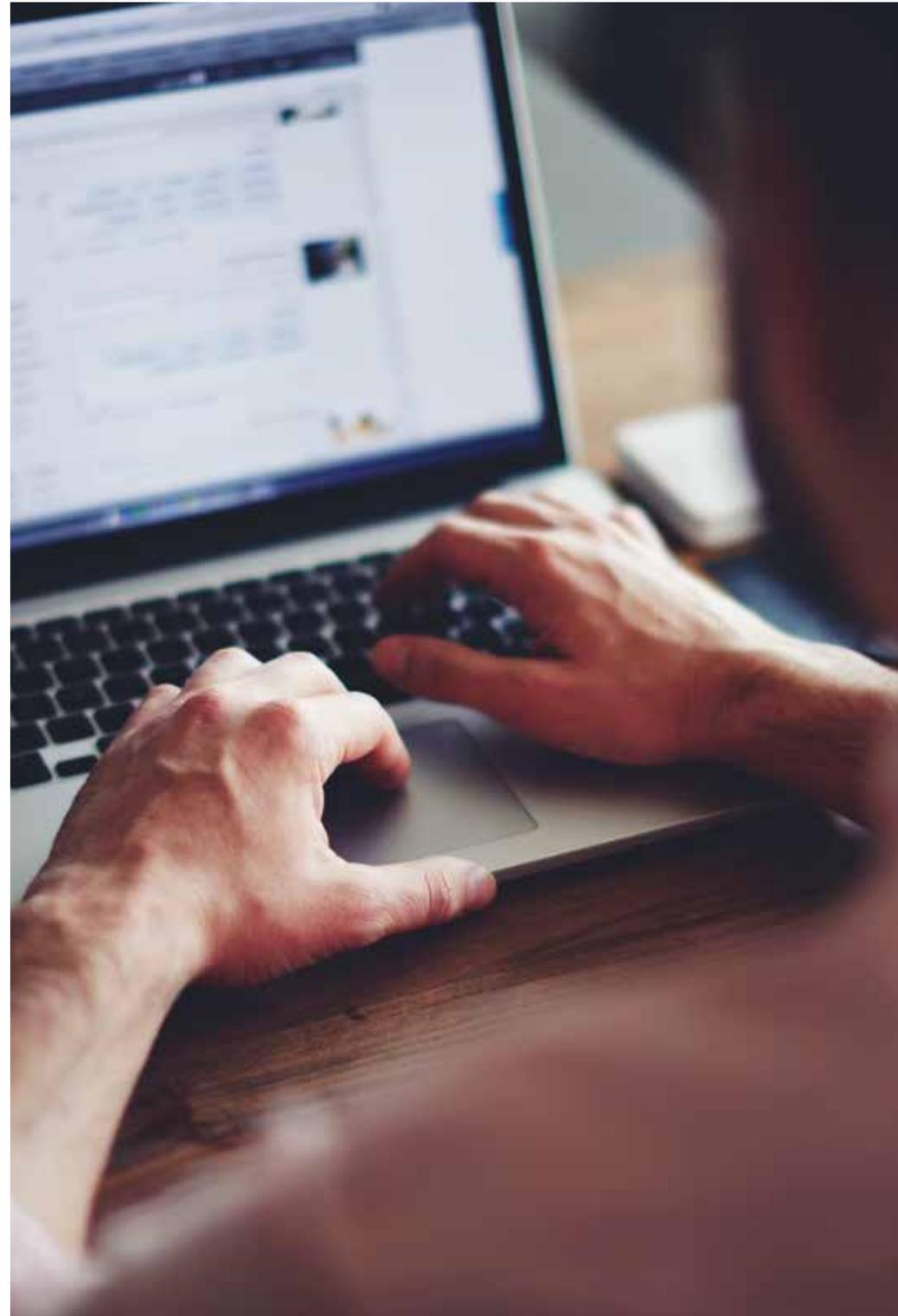
APPDYNAMICS

Contents

- The Quest for Optimal SharePoint Performance **3**
- The Whys and Hows of Monitoring SharePoint **9**
- The Anatomy of SharePoint..... **13**
- The Top 8 Challenges in Troubleshooting SharePoint..... **18**
- Configuring SharePoint With AppDynamics..... **23**

01

The Quest for Optimal SharePoint Performance



Who Should Read This Book

The primary intended audience for this document is any expert in professional services who needs to assure the health of SharePoint and systems connected to it.

If you're a systems architect, you can gain understanding of SharePoint components and how other applications can take advantage of SharePoint. If you're an independent consultant, you'll learn about the elements of comprehensive coverage and total visibility into operations with prebuilt monitoring configurations.

Sharing This eBook With Users

Everything in this eBook is based on real-world examples and configurations where AppDynamics was deployed to monitor SharePoint. There's no private or identifying information here, but it is not designed for the average SharePoint user. The level of detail can be overwhelming and not very useful for non-technical colleagues.

This book can be used to help guide users through specific problems, but make sure that you work through those issues first and do not confuse the user with topics that take more specialized technical knowledge and experience.

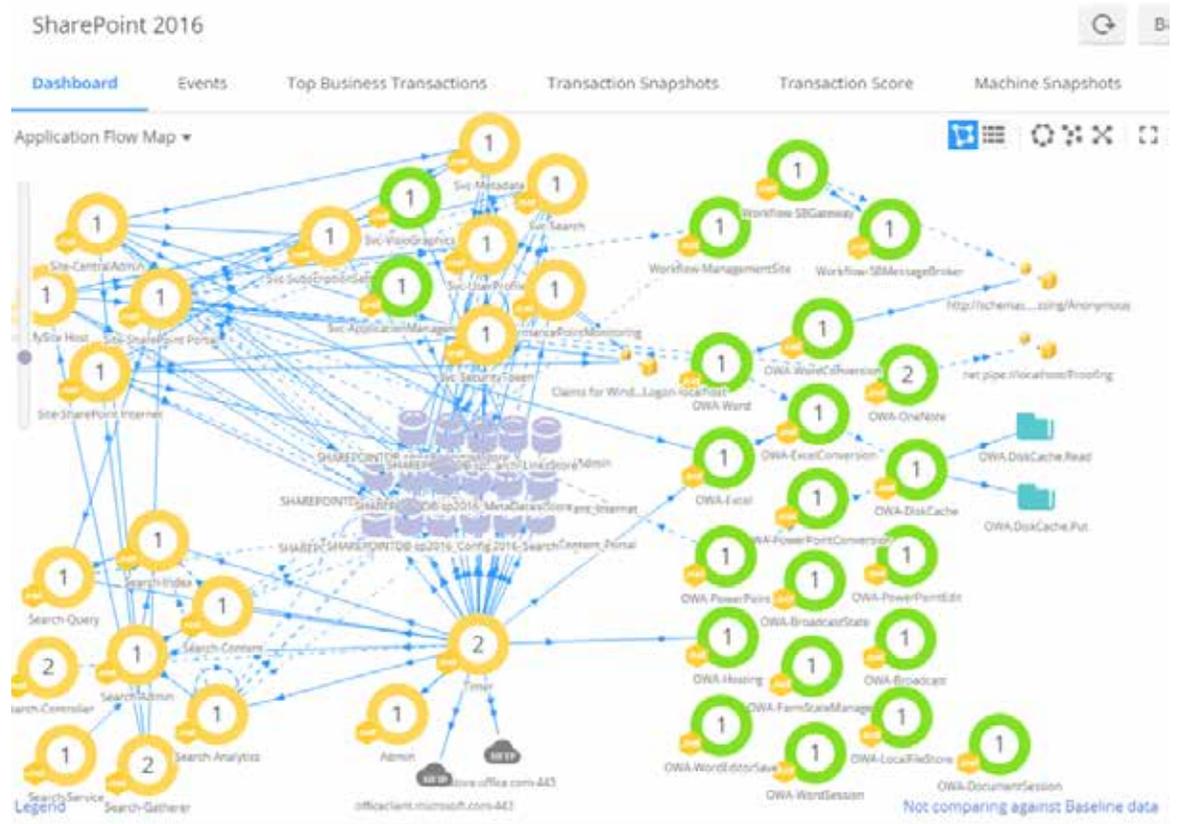
How Much Do You Know About SharePoint?

For the vast majority of SharePoint users, that question is impossible to answer. They often repeat the marketing description that SharePoint is a powerful and extensible platform for business management and collaboration.

Project managers might tell you that they know SharePoint is deployed in a large percentage of medium to large-sized enterprises around the world, in on-premises versions or hosted in the cloud.

Most likely, users will simply say that SharePoint is a very complex document sharing system that the boss expects them to use.

This eBook will help you see SharePoint performance from the end user's point of view and to make life easier for them by keeping it operating at maximum potential. Don't be surprised if users can't easily tell the difference between normal processing times and a total breakdown in cross-functional communications.



If you're one of the fortunate few responsible for seeing that SharePoint turns in peak performance, you may experience one of the following three challenges in understanding the platform right off the bat:

1. SharePoint is astonishingly complex software with 50+ executables, various services dependencies, and more databases than you might expect.
2. Evaluating and acting upon the health of any specific component within SharePoint requires interpretation of PerfMon counters that will drive you and your team to dig through mounds of SharePoint ULS logs.
3. The freedom of customization that characterizes SharePoint implementations is both a top-selling feature of the software and the central cause of headaches for software support experts. Customer extensions connected with the SharePoint ecosystem frequently cause performance characteristics to change in unpredictable ways.

When that happens to you, remember to step back, take a deep breath, and retrieve this eBook.

When a user comes to you with a SharePoint malfunction, your first instinct will probably be to investigate whether the user is in need of a training refresher. That's often true, because it's hard for users to know when SharePoint is not working as intended.

Don't be surprised if it's difficult even for you to be certain when something's gone wrong in some cases because:

It's not easy to trace a process of execution in SharePoint from end-to-end.

- Some types of services' failures are not obvious immediately, but they do impact other services in a cascade of failures.
- Sometimes a flurry of issue reports come in from users experiencing the exact same breakdown in different ways. In those cases, it becomes difficult to stay ahead of the issue tickets, as users describe malfunctions in conflicting ways.

In this maelstrom of uncertainty, AppDynamics is your first line of defense. AppDynamics can monitor most of SharePoint components with fine-grain precision using a mixture of .NET APM agents, EUM agents, analytics agents, and machine agents.

An Overview of SharePoint

According to Microsoft, the official description of SharePoint is that, "SharePoint is a collaboration environment that organizations of all sizes can use to increase the efficiency of business processes."

Over the past few years, SharePoint has grown to be incredibly pervasive across the enterprise. It's hard to find an organization that isn't using it or hasn't used it.

There's a stickiness to SharePoint, also. Once embedded and actively used, it becomes difficult to replace. SharePoint pulls lots of other Microsoft components, like Windows, SQL, Office Web Apps, and Office licenses. Microsoft loves selling it and has been extremely successful in doing so.

There's a very healthy third-party vendor ecosystem that has developed around SharePoint, including:

- Extensions for business process automation
- Custom web parts and applications
- Extensions for storage
- Monitoring and management extensions
- On-premises version of SharePoint hosted and managed by a variety of partners
- Custom software developers/system integrators

Workloads of SharePoint include:

- Collaboration on documents and business processes
- Secured intranets and extranets
- Search for documents and business data
- Anonymous and secured public websites, including web content management and content publishing
- Business process automation
- Records management
- Social integration

Business Process Automation

One of the strongest uses of SharePoint is business process automation. Workflows are built around documents in libraries or items in lists. The workflow can be either declarative or code-based, and can be very long-running. Declarative workflows are relatively easy to develop without any knowledge of the code and are frequently created by power users.

Editing of declarative workflows is done through SharePoint Designer. Code-based workflows are created using Visual Studio and deployed to the SharePoint farm using extension tools.

For SharePoint 2010, the in-box workflow technology was based on Workflow Foundation 1.0/2.0 and tightly integrated with SharePoint. That ended with the SharePoint 2013 upgrade, when Microsoft rewrote the workflow engine.

The new engine was based on Workflow Foundation 3.0. Unlike the 2010 workflow engine, the 2013 version runs on a separate server from the main SharePoint farm. The services connect to SharePoint via public and private APIs to obtain and manipulate data, as well as their own databases.

Nintex Workflow also offers extensions for SharePoint Workflow, in both 2010 and 2013. It is a very well-respected solution and in some cases runs people's businesses on top of SharePoint.

Another popular and well-reviewed option for Workflow platform extensions is K2 BlackPearl.

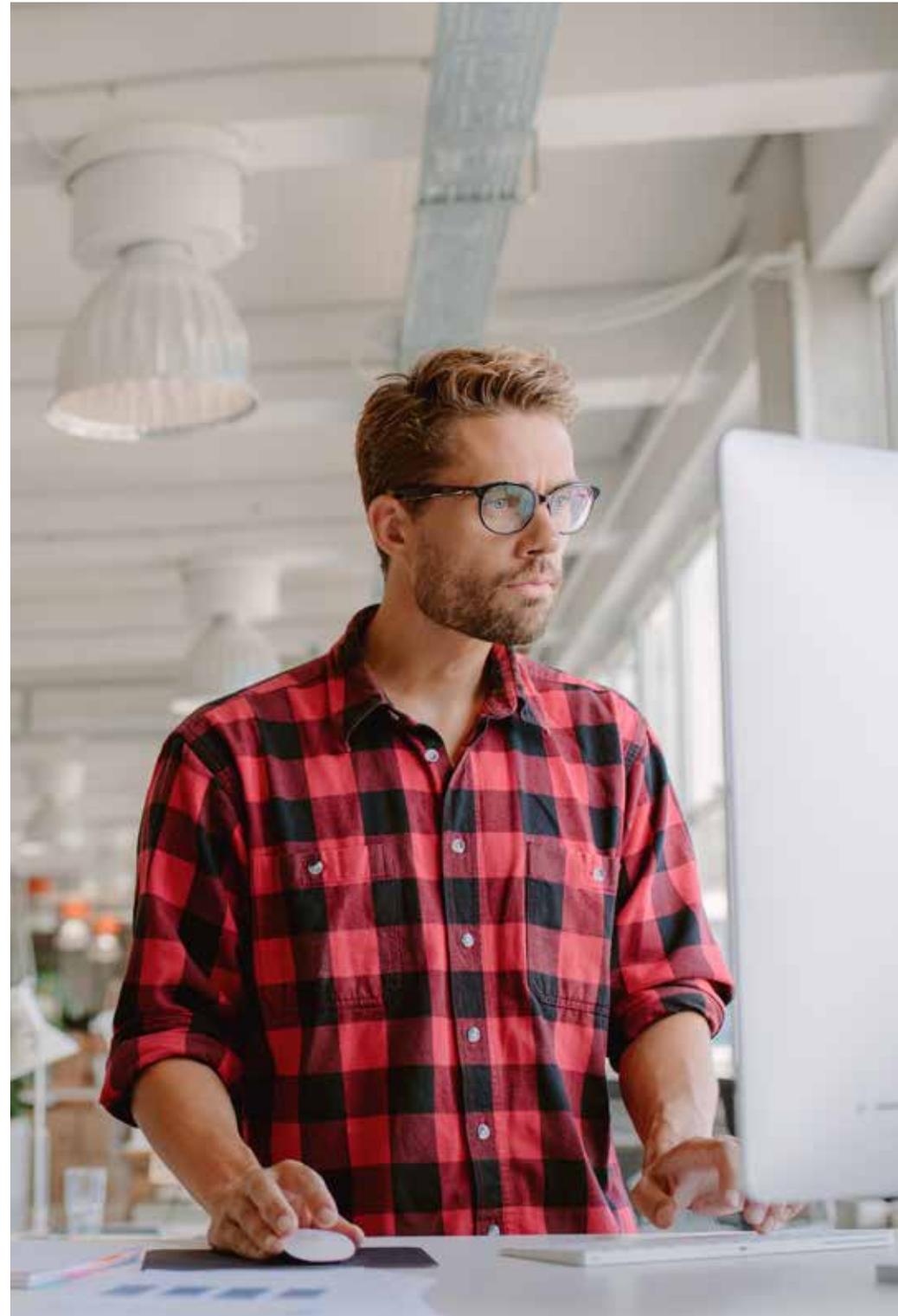
Better Together

It's critical for any business running SharePoint to achieve optimal performance and availability of their applications. Business operations must be able to assure their internal and external teams full access to the contents and collaborative services that SharePoint offers. To do that, administrators need a simple but robust application performance monitoring solution with total visibility across the SharePoint environment.

In the next chapter, you'll find out more about how SharePoint monitoring and AppDynamics work together to bring you all the details you need for complex tasks like end user monitoring, log analytics, and server support.

02

The Whys and Hows of Monitoring SharePoint



There's no question that SharePoint is a very large beast. In the old days, it would be labeled as a "megafauna." SharePoint has a well-deserved reputation for giving support headaches due to all the many interdependent moving parts that we will examine in greater detail in the next chapter. Before you get there, however, you need to make sure you have a good foundation of why and how SharePoint and AppDynamics work better together.

Over time, SharePoint has become even more resource-intensive due to its many capabilities and customized extensions. Although internally it has plenty of telemetry, SharePoint is mostly log-driven, which can be very difficult to interpret. It requires extensive tools and training to bring all of SharePoint's specialized systems under control.

What AppDynamics Does for SharePoint

AppDynamics can help operations understand this megafauna with scientific precision. For anyone tasked with developing on this platform, AppDynamics is here to streamline the code and the whole SDLC.

Once you understand how the holistic system functions and you can visualize it in the wild, the activity monitoring can leverage baselines and deviations from them with Health Rules to alert everyone on your team about issues. AppDynamics and SharePoint work better together.

APM solutions help teams understand what's happening on the end user page and document navigation/browsing. They are built to break up process flows by the individual web application.

Document collaboration is key to SharePoint workloads. That's another reason to deploy AppDynamics. APM solutions clearly lay out office document rendering behavior and highlight any issues that arise along the way. When Office Web Apps (OWA) break, the messages they spit out are normally pretty cryptic. AppDynamics clarifies that.

The final reason you'll want an APM solution supporting SharePoint is to get more details on search query usage. SharePoint has some usage analytics on search, but you'll get more detailed results faster when the two of them work together.

This is an important point because AppDynamics is not meant to be a replacement for native monitoring tools. Rather, it's a sophisticated tool for deeper investigation and simplification of issue resolution.

AppDynamics operates in tandem with standard SharePoint monitoring tasks such as monitoring SharePoint Event Logs, reviewing Timer Jobs in Central Administration, regularly reviewing great volumes of ULS Logs, and looking at Usage Logs. Many find that the most important task from a user experience perspective is in respecting and remediating Health Rule violations.

You can investigate SharePoint native monitoring documents on Microsoft's TechNet.

Note that AppDynamics is also not a replacement for the Systems Center Operations Management (SCOM) pack or a well-planned Tivoli implementation, particularly when just interpreting PerfMon counters. SharePoint Server 2013 integrates well with SCOM, which is very useful but still optional. Not everyone running SharePoint will need the SCOM pack, but they will certainly need some kind of APM solution.

The Challenge of UI Customization

The vast majority of SharePoint users will access their most common workflows through a web browser. A few users will access it via SharePoint Designer or indirectly through an Office client like Word, Excel, or Outlook.

One of the big features companies love about SharePoint is how easily the UI can be customized in order to meet the unique business needs of each user group. As a result, the UI is almost always customized and personalized, which frequently means SharePoint is more likely to break down or become very slow in certain types of browsers.

You have to be careful digging into the UI because SharePoint generates an incredibly complex mass of various scripts. Based on countless variables like the browser type, browser version, enabled features, and more, the scripts are served up to end users in endless variations.

You also need to be aware that the pages are very heavy with AJAX. Many customers go further and extend SharePoint with their own scripts. For instance, it's common to use JQuery or Bootstrap to make the UI more responsive, which is going to further complicate the process of tracking down performance issues on these pages.

How to Approach End User Monitoring (EUM)

AppDynamics helps operations teams understand and manage page complexity from many different angles. It can also show differences in the behavior of users at various remote sites.

Fortunately, the AppDynamics end-user monitoring (EUM) capabilities display browser and mobile application metrics with incredible detail. These tools are also useful for tracking down web clients that generate resource-intensive calls, both of which can slow down the entire Business Transaction. With AppDynamics, you have full visibility into what's taking place behind the scenes.

AppDynamics EUM script injection is straightforward and there are multiple methods that are supported. Automatic injection works beautifully for all pages. However, some security-conscious teams may not feel comfortable about injecting scripts site-wide. If so, you can also choose to inject them using self-hosted EUM scripts, either in the SharePoint libraries or via deployment to the Web Application-wide _layouts virtual directory.

Transaction Analytics and APM

Most but not all business transactions are clarified with context via HTTP and MIDC data collectors. This helps both traditional APM and transaction analytics. The package introduces a great deal of data collectors that can enable powerful troubleshooting tools.

Some SharePoint users also use AppDynamics to analyze usage of various sites in comparison to each other. Anytime a business using SharePoint wants to dig deeper into usage stats, AppDynamics can fulfill their knowledge quest and easily integrate with third-party content analysis solutions for SharePoint content repartitioning.

Reading Logs

SharePoint logging is done with Universal Logging Service (ULS) logs. These are large files produced by the SharePoint platform on each server. You may find that you are spending a great deal of support time in dealing with SharePoint ULS logs and interpreting their contents. The records in the logs are correlated using a Correlation ID GUID that travels across multiple log areas and executables.

As you may suspect, these logs can and do grow to be vastly unmanageable. Busy SharePoint farms can produce gigabytes of logs a day. The log contents may be difficult to interpret because they vary widely based on slight variations in the customer workload.

Keeping the Server Healthy

The Microsoft Technet site says that server requirements for SharePoint are relatively modest. Don't underestimate what you need. Make room for growth and customizations. At minimum, double the recommended server resources to get decent performance and avoid pagefile churn.

It's important to note that the web and application front-ends don't usually keep state or store data. It is possible to turn on, but it's rare that anyone ever does. Your file system speed or size will not have significant importance unless you're caching blob content, which is relatively rare.

Some search server components have their content index stored on file systems, which can be big, but it doesn't have to be extremely fast. Your .NET Machine Agent already handles tasks like watching the CPU and memory metrics on web and application servers. If you want to keep track of more data about your servers, you can use the AppDynamics APM agent to surface PerfMon counters.

Due to its function in document management, collaboration facilitation, and version control, SharePoint is database-intensive. All production farms always have dedicated SQL Server hosts because that's where most of the heavy lifting is done. Hosting can be a struggle as databases expand rapidly to keep up with hosted content growth.

Database Agent can see inside of SQL Server for SharePoint just fine and should be offered to prospective customers as part of the package. Because most enterprises run SQL Server for SharePoint in Integrated Authentication mode, you will need to configure DB Agent access using Integrated Windows Authentication (IWA).

Assembling the Component Pieces of SharePoint

Whenever you encounter an issue with resource utilization or latency, you want to discover which components are pulling on those resources and make sure that they are operating at full capacity before you allocate more resources.

A good example would be when you see a web server slowing down with too much static content. You probably want to check log settings of the IIS and the modules that it loads. Log levels often get increased for troubleshooting and never pushed back for maximum performance.

In the next chapter, we'll examine the major components inside SharePoint, which includes vitals systems related to:

- Web Front End Server
- Office Web Apps
- Farm Topologies
- Application Pools
- Cross-Tier Correlation via HTTP, WCF, and Web Services
- User Profile Synchronization
- Many more

You'll be able to see how they fit together and how AppDynamics provides deep code-level diagnostics to pinpoint slow business transactions as they flow through the SharePoint architecture. You'll refer to Chapter 3 often to resolve performance issues faster and review performance details at the transaction level.

03

The Anatomy of SharePoint



Now it's time to look deeper into SharePoint's major components and how they work together. Those component categories are:

- Farm components
- Web components
- Central Admin and Databases
- Web services
- SharePoint Designer and OWA

SharePoint Technical Diagrams give you detailed, large-format visuals that can help you get oriented. Another great place to start is the Microsoft SharePoint Server 2013 architecture diagram. If you want a more traditional topology for SharePoint 2013, Microsoft can give you those as well.

Farm Components

SharePoint Farm – A logical combination of various servers that work together to provide content and services.

Web Front End Server – Fast, lightweight server that responds to user requests for web pages. All web servers in a farm are mirrors of each other and are load balanced. It hosts web pages, web services, and web parts that are necessary to process requests served by the farm. It directs requests to the appropriate application servers.

Application Server Role – Provides the service features of SharePoint products and technologies. An application server often provides all or a subset of service features. Multiple redundant application servers can be load balanced.

In many farms, all services will run on two identically configured application servers for redundancy.

Database Server Role – Stores content and service data. All databases can be assigned to one database server. Databases can be spread across multiple servers. All databases can be clustered or mirrored for failover protection. In a small farm environment, all databases can be deployed to a single server. In larger environments, you can group databases by roles and deploy these to multiple database servers.

Office Web Apps Server – This serves multiple SharePoint Server farms for viewing and editing. You can view files from Exchange Server, Microsoft Lync, etc. This integrates with URL-accessible file servers. Optional Workflow Manager Server enables SharePoint 2013 Workflows.

Active Directory – SharePoint depends on this for user source and authentication, although it can run anonymously.

Farm Topologies – SharePoint is flexible in sizing of the farm and configuration of components. A one-server farm can easily handle up to 100 users, and a two-server farm – with the second server for the database – can handle up to 10,000 users. Six to eight server farms are common at most mid to large-sized companies.

Web Components

IIS Application – An application enabled for SharePoint uses an ASP.NET engine to register the SharePoint content handler, and can host a variety of site collections, each of which contains a multitude of websites. Users can access SharePoint sites anonymously or with various forms of authentication. Content is served from content databases in SQL Server using a mixture of file system-hosted and SQL Server-hosted ASP.NET pages, ASP.NET page handlers, and web services. Because of how AppDynamics plugs into processes, the unit of monitoring is IIS Web Application, a.k.a. SharePoint Content Web Application.

Application Pool – Provides settings such as user account, .NET Framework version, recycling, memory thresholds, and others for IIS Web Application. The vast majority of customers will run SharePoint Application Pools using domain accounts.

Web Applications – These are created wherever the default IIS wwwroot is. Central Administration is a special site that always gets created with random numbers corresponding to its initial auto-assigned port. Find its location using these PowerShell commands:

- Import-Module WebAdministration
- Get-ChildItem IIS:\Sites

Authenticated Site – All SharePoint sites are created using SharePoint Central Administration UI or PowerShell commands. The authentication is configured using Central Administration or PowerShell. For sites with Claims Authentication, “_login” virtual directory will be used to authenticate. For sites with forms authentications, customers may be adding their own pages to the site.

Extended Sites – In some instances, SharePoint can extend coverage from an existing SharePoint Web Application to a new or existing IIS Web Application. This is rare, but when it happens, it is like a clone of an existing site, except they are linked.

Content Site – This is the main way users interact with SharePoint. It is a set of ASP.NET pages and various services that uses the SharePoint content handler to translate requests to various database and service calls.

My Site – This is the sub-type of a content site that hosts special instances of SharePoint Website for individual users’ home pages. If configured, each user in the organization gets an instance that they can customize. Think LinkedIn or Facebook-style profile, with some light document hosting. My Site is typically built with a user profile that is fed from Active Directory and can synchronize data back to it.

Virtual Directories (_layouts, _vti_bin) – SharePoint Web Applications are characterized by various virtual directories, which map to the “C:\Program Files\Common Files\microsoft shared\Web Server Extensions\###” folder where ## is 14 for SharePoint 2010, 15 for SharePoint 2015, and 16 for SharePoint 2016.

Managed Path – This is a location within a web application in which you can have site collections. When you create a web application, there are two managed paths created with it. The first managed path is called the Root “/” path of explicit inclusion type. The second is called “sites” with wildcard inclusion path.

A typical SharePoint URL could be: “http://company.com/sites/Sales/apac/,” where:

http://company.com – Web Application

Sites – Managed path Sales – Site collection apac – Sub-site

Two types of managed paths are:

A wildcard inclusion allows you to append multiple site collections to the path that you specify. For example, if you add “/engineering” as a wildcard inclusion off of your root site of “http://contoso,” then you’ll be able to create multiple site collections off of “http://contoso/engineering.” An explicit inclusion allows you to create a single site collection with the specified address. For example, if you add “/finance” as an explicit inclusion off of your root site of “http://contoso,” then you’ll be able to create a single site collection with the address “http://contoso/finance.”

Almost all customers will have "teams" and "sites" managed paths, and many will have more. For more, see "Define managed paths in SharePoint Server 2013."

Site Collections and Websites – A site collection is made up of one top-level site and all sites below it. From an architecture standpoint, all the content of a site collection must be stored in a single content database. You cannot have a site collection's content spread out across multiple content databases. Content databases scale with your infrastructure capacity, so site collections can share a content database.

Although a site collection can exist in only one content database, one content database can host the content for multiple site collections. Similarly, any given SharePoint 2013 site can only exist in one site collection, but a site collection can host a multitude of sites. A site cannot exist outside of a site collection.

Central Admin and Databases

Central Administration Web Application is always called "SharePoint Central Administration v4." The site is created during Farm configuration at the port of the user's choosing. Central Administration site is frequently called "SCA." On SharePoint servers, there is a Start Menu shortcut called SharePoint Central Administration that will open it.

SharePoint uses SQL Server databases. The most important database is the Farm Configuration database. Everyone uses it. Almost every component has its own database. In the base configuration, 15-17 databases are common. In complex environments, it is not uncommon to reach 25+ databases. Some databases are small and used infrequently. Others can grow huge and require a lot of database resources, especially the content databases.

Web Services

Hosted Service Names – SharePoint has a multitude of ASMX Web and WCF Services that are leveraged by UI and server-side components. They are hosted under "SharePoint Web Services" IIS Web Application under the most unintuitive IIS Virtual Directory naming schemes using GUIDs that are unique to each Farm.

Although they seem random, The GUIDs used as IIS Virtual Directory names are actually meaningful. Normally, the name of the IIS Virtual Directory is the GUID of the underlying service as known to SharePoint with dashes stripped out. It's possible to get those services IDs by running the following SharePoint PowerShell command:

```
Get-SPServiceApplication | Select DisplayName, ID | Sort ID
```

Having these services mandates using IIS Manual Tier mapping in AppDynamics .NET Agent config.xml. Miscellaneous services include:

- Search Admin and Search Service invoked by UI components
- Metadata
- User Profile
- Topology
- SharePoint Time Service (SPTimerV4)
- Unified Logging Service (SPTraceV4)
- User Sync (for SharePoint 2010 and 2013)
- User Profile Synchronization (for SharePoint 2016)
- AppFabric Caching Service
- Search Services

SharePoint Designer and OWA

SharePoint Designer – This end user tool is for customizing SharePoint websites and building workflows. It is part of the Office suite. It is an essential tool for developers and business process designers.

It uses SharePoint Web Services, Client Object Model, and old-school FrontPage RPC calls to talk to SharePoint.

Office Web Apps (OWA) – The server delivers browser-based versions of Office apps in an on-premises environment. OWA components were originally installed into SharePoint 2010. In SharePoint 2013, OWA must run on separate boxes. OWA is horizontally scalable. OWA does not use any databases. All communications between OWA and SharePoint happen via HTTP calls and WCF services.

Office Online Server – This is the new name for OWA in SharePoint 2013. New features were added but the architecture is the same. Just like SharePoint itself, components of OWA in 2013 and OOS in 2016 produce large amount of ULS logs.

See: Plan Office Web Apps Server – <https://technet.microsoft.com/en-us/library/jj219435.aspx>

OWA IIS Websites – These components that provide rendering services run under IIS as a series of ASP.NET pages, ASP.NET page handlers, ASMX Web Services, and WCF Services.

Identifying Issues and Fixing Them

The next chapter gives practical steps for tracing down the components involved in the most common issues, as well as suggestions on how to approach resolution.

Whether you're picking your way through complex customizations, overly chatty browser calls from inefficient batching, or just trying to follow the flow through a dizzying array of moving parts, you'll get a jump on the problem in the pages ahead.

04

The Top 8 Challenges in Troubleshooting SharePoint



This is where all of the components specified in the previous chapters come together. As you might expect, the more customizations there are, the more uncertainty enters the system and the greater the chance for inefficient code.

It will all work out as long as you have a continuous source of monitoring in place to fix errors as they happen or to fine-tune performance so errors don't happen in the first place. You will have to keep track of multiple business transactions simultaneously in order to identify poorly performing components like SharePoint sites or SQL queries.

As you work to maximize system resource utilization, eventually your team is going to experience overloaded SharePoint services or see their other network applications impacted.

This is a look at some of the most common SharePoint issues that have been reported and what AppDynamics does to put you on the right path to resolution.

1. Overly Complex UI Customizations

When SharePoint 2013 hit the market, enterprise clients were excited about all the new possible customizations to the UI, making it more user-friendly, faster, and more agile.

The price to pay came on the back end, as business operations teams had to support complicated customizations that tended to use system resources inefficiently.

For example, many times a developer will build a complex web part that does a great many actions on the server. These usually end up becoming too expensive in terms of server resource allocation over the long run. If your front end is getting a bunch of data via calls to a server object model, those typically take a long time to render. You can see exactly what is happening, and what should be happening, with AppDynamics.

2. Impacts on UI Performance Over a Poor Network

Just rendering the SharePoint UI takes a huge amount of scripts and DHTML. In the last chapter, you learned about how the UI in SharePoint is built up from web parts or units of display that run in context with the page. These can be repositioned as needed in the web part regions defined by master page and page layouts. The point was to take some of the pressure off developers by allowing it to be done in the browser by non-technical team members with little to no coding skills. In many cases, users can also seek out and add new pre-built web parts from various vendors in the market.

That's great for the user, but there's a significant cost on the server side. UI can also be customized using Search Web Parts that execute their own in-line searches and provide results for display. Extending them means the developer needs to become proficient at the obscure type of markup and JavaScript required by SharePoint.

All of those communications add up quickly, making the entire system very susceptible to degradation over slower networks or when poor caching is done by CDNs. In this case, the AppDynamics EUM tools show you exactly where the slow-downs are originating so you can start moving in the right direction.

3. Overuse of Client Object Model

Web parts are the culprit in this case as well. Developers frequently build a customized web part that renders things by calls to the Client SharePoint Object Model (CSOM) via the external API. The CSOM is a set of APIs that's callable from JavaScript, .NET code, SilverLight, etc. The APIs conduct messages and communications to the WCF service endpoint, providing translation from returned results.

There is also a REST endpoint for CSOM that returns XML or JSON. With CSOM, any application outside of SharePoint can get data out and put data back into SharePoint.

As you can see, it's easy for developers to use CSOM inefficiently. SharePoint supports batching, but if it's not handled efficiently, you will see too many calls hitting the server. This is another instance where the AppDynamics EUM clearly shows where the problem originates.

4. Custom Event Handlers

More freedom means more problems. That's certainly the case with custom event handlers. Expect these to handle events on the server in a way that is either too buggy or too slow.

A common problem also crops up in registering a Remote Event Receiver. It's automatic for SharePoint to add the App ID to the Event Receiver's registration information. When you want to deactivate a specific Event Receiver, you can only do that with the same app that registered it. If you attempt to remove an Event Receiver registered by an app with another ID, it throws back a "Server Unauthorized Access" exception.

AppDynamics gives you full visibility into malfunctions of your custom event handlers.

5. Bad Database Configuration

As discussed earlier, the database tends to do the heavy lifting, and that only gets more intense as your documents grow. Even when you have assigned a server or a set of servers to the database alone, if your database administrators provision log and data files are on the same disks, you can expect to eventually see some performance issues.

AppDynamics will bring this issue to light because you will see query times start to climb. You will also see indications of an inefficient database setup in how applications are impacted by indexing and reporting services. You can be alerted when AppDynamics auto-detects architecture changes in production.

6. Underpowered Web and App Boxes

Not enough operations teams allocate adequate resources to the Web Front End (WFE) server. This is the server that will handle web page requests from users, process those requests, and then send back the data. They process HTTP/S traffic and coordinate traffic with the IIS. On some of the smaller enterprise topologies, the WFE and the application server functions are combined.

As the company and the SharePoint implementation grows, that becomes a problem. This is because WFEs need a significant amount of memory and a decent amount of CPU. Meanwhile, the application servers need even more memory and CPU power. When they aren't adequately supplied, they starve each other out.

After you set up Health Rules in AppDynamics, you will be able to tell instantly where you are using too much CPU power and when you may be getting close to hitting full memory utilization.

7. Managed Metadata Service

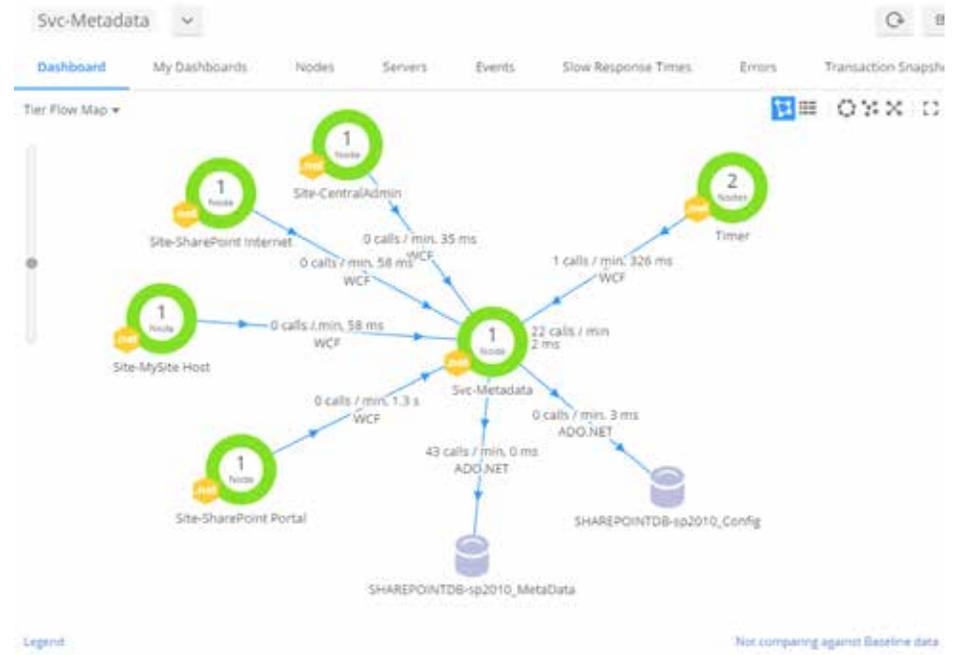
Some operations teams consider Managed Metadata Service to be one of the least reliable components in SharePoint and look there early in the troubleshooting process. SharePoint has many ASMX web and WCF Services leveraged by UI and server-side components. You will find them hosted under the "SharePoint Web Services" IIS Web Application.

The naming schemes have gotten more reasonable under SharePoint 2016, but if you are using an older version, you will have to compare GUIDs of SharePoint Services as known to SharePoint to GUIDs of IIS Virtual Directories in order to tell services apart. Having these services mandates using IIS Manual Tier mapping in AppDynamics .NET Agent config.xml, which you will see in the next chapter.

In the following illustration, you can see a flow map of dependencies of a typical Managed Metadata SharePoint web service.

The Metadata Service in the center gets queried many, many times during page building.

It should come as no surprise that when it breaks, typically returning 500 errors on queries, it degrades the UI experience for anyone who is browsing the company sites or using a faceted search. Again, this is an error that is probably not obvious from the front end, but it can be easily traced using AppDynamics.



8. Understanding All the Moving Pieces

You've looked over the components in the previous chapter and you probably have some experience working under the hood with SharePoint by now. However, you may not realize how quickly things can change when it comes to common business transactions.

To get a better feel for how this works in a real world example, take a look at a sample "Pages.Search" business transaction.

This business transaction wraps search results page invocation. "OSSSearchResults.aspx" is a built-in Team Site page that is part of every site collection and is used by default, unless overridden by the Site Collection Administrator.

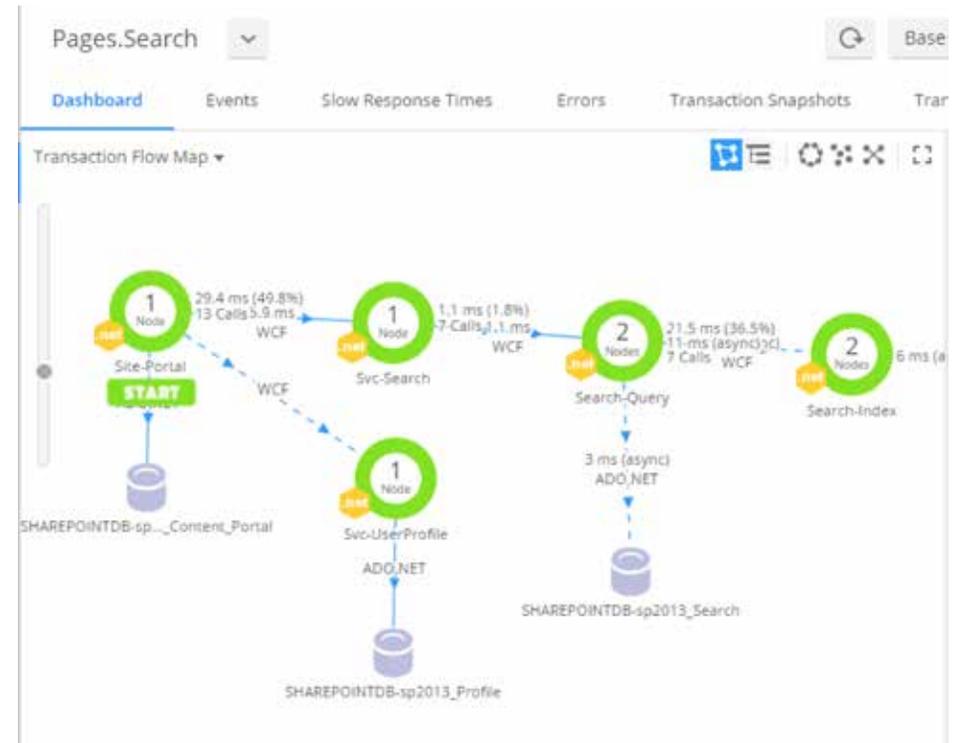
When overridden, search results are provided by instance of the Search Center web site, which typically has "results.aspx", "peoplereults.aspx", and so on.

The result pages can also update their findings in AJAX fashion, which this rule sort of captures when there is a partial postback. You can see that there are many component interactions involved just in making a simple search.

Without AppDynamics, tracing this kind of business transaction would be frustrating, to say the least.

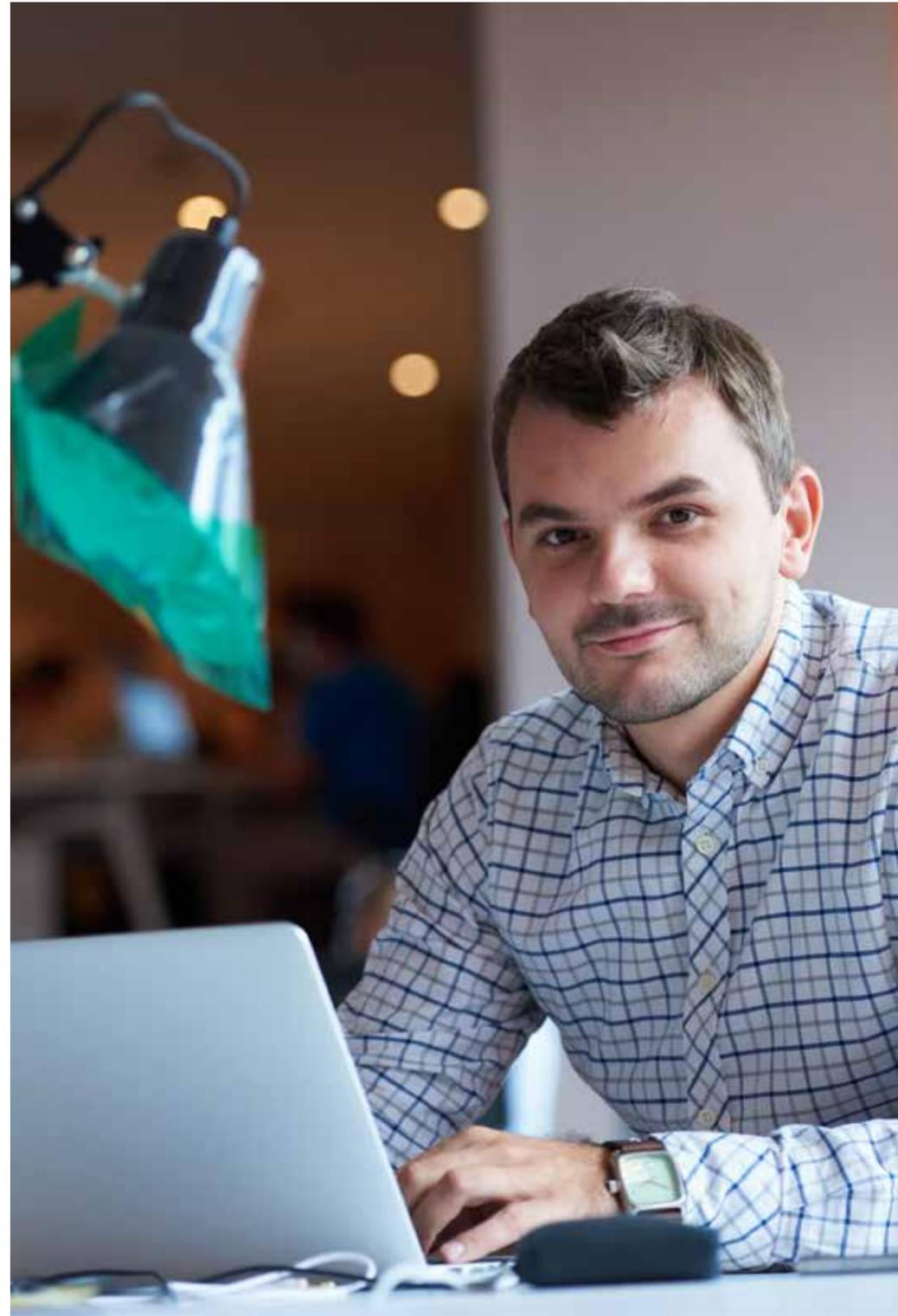
Setting Up for Business

In the final chapter, you'll walk through setting up AppDynamics to start monitoring the performance of your SharePoint implementation. You'll prepare the AppDynamics application and setup the config.xml file with your own unique set of parameters. Finally, you'll have a chance to review actual flow maps and start building up a visual databank of what normal operations look like. From that point on, it's your show.



05

Configuring SharePoint With AppDynamics



By now you should have a good understanding of how all the pieces work as a whole and be ready to get started. The first step will be to obtain a copy of the template config.xml file from AppDynamics.

You won't be able to use it directly, though. You must first convert it using a PowerShell script.

Converting the Template to the Working File

Next, you will update the template config.xml to include Content Web Applications and Web Service Application GUIDs.

Follow these steps to create real config.xml from template config.xml:

- Log in to a Web Front End member of the SharePoint farm as administrator
- Open PowerShell in Elevated mode
- Copy ConfigureSharePointForAppDynamics.ps1 and SharePoint.Template.config.xml to the server somewhere
- Run the following command from where you created the files:

```
\ConfigureSharePointForAppDynamics.ps1
```

Or, explicitly:

```
.\ConfigureSharePointForAppDynamics.ps1 -InputFile ".\SharePoint.Template.config.xml"
```

The resulting config.xml file is created by the script for you and includes a full list of monitored sites and SharePoint websites' GUIDs mapped to the right tiers. This initial file is setup to accommodate three instances of each of the Search Components. That's why NodeRunner.exe entries are repeated three times. If you have more than three instances/partitions of each, you will need to contact AppDynamics support before continuing.

Preparing the AppDynamics Controller

Add controller host, port, application name, account, and account key at the top of the resulting config.xml so you can connect to the controller.

To set up the controller, you will preconfigure the application with the rules provided in the Config Exporter rule files. You will need a copy of Config Exporter, version 4.2.5 or later. You can get this file from AppDynamics. The JSON files you need will be attached to this document.

Increase the number of available Business Transactions for detection to at least 300. This is a controller-wide domain1.xml setting that requires controller restart. If using AppDynamics SaaS environment, only dedicated controllers allow that.

Now you are ready to create a new AppDynamics Application in Controller.

Here are the rules to import with their source files:

- Import Node properties rules from the appagent-properties.json file
- Import Business Transaction rules from the entry-points.json file
- Import Business Transaction Exclude rules from the transaction-detection.json file
- Import Backend naming rules from the backend-detection.json file
- Import Custom Exit Point naming rules from the exit-points.json file
- Import Data Collectors rules from the data-collectors.json file
- Import End User Monitoring rules from the eum-configs.json file

Installing Agents on SharePoint Farm Servers

Deploy agents to members of SharePoint farm:

- Web Front Ends
- Application Servers
- Search Servers (type of Application Server)

Now you will choose the .NET agent version. This configuration has been tested with .NET Agent version 4.2.8. Choose 4.3.(latest version) of the 64-bit agent. Be sure you do not deploy this with 4.1.x .NET Agent.

Manual Installation

- Install Agent from MSI wizard
- Run configuration utility and complete Log Permissions screen
- Do not go any further – dismiss configuration utility
- Copy config.xml created previously using the template creation script to %ProgramData%\AppDynamics\DotNetAgent\Config
- Restart AppDynamics.Agent.Coordinator (AppDynamics.Agent.Coordinator_service) service
- Ensure Machine Agent registers with AppDynamics Controller using this server's name
- Restart monitored processes. Run iisreset to restart all IIS web-sites:

```
iisreset /noforce
```

- Restart monitored services if they are running and are not disabled – SharePoint Timer Service and the SharePoint Administration:

```
net stop SPTimerv4
net start SPTimerv4
net stop SPAdminV4
net start SPAdminV4
```

Restart Monitored Services on SharePoint 2010

- SharePoint Server Search 14
- Forefront Identity Manager Service

```
net stop OSearch14
net start OSearch14
net stop FIMService
net start FIMService
```

Restart Monitored Services on SharePoint 2013

- SharePoint Server Search 15
- Forefront Identity Manager Service
- SharePoint Search Host Controller

```
net stop OSearch15
net start OSearch15
net stop FIMService
net start FIMService
net stop SPSearchHostController
net start SPSearchHostController
```

Restart Monitored Services on SharePoint 2016

- SharePoint Server Search 16
- Forefront Identity Manager Service
- SharePoint Search Host Controller

```
net stop OSearch16
net start OSearch16
net stop SPSearchHostController
net start SPSearchHostController
```

For each service, confirm that the agents register with AppDynamics Controller.

Alternatively, use the AppDynamics PowerShell deployment extension to install everything remotely.

Note that you will need to convert the config.xml file created previously from a template into a template acceptable for unattended installation as described in the extension documentation.

End User Monitoring Pages and AJAX Requests

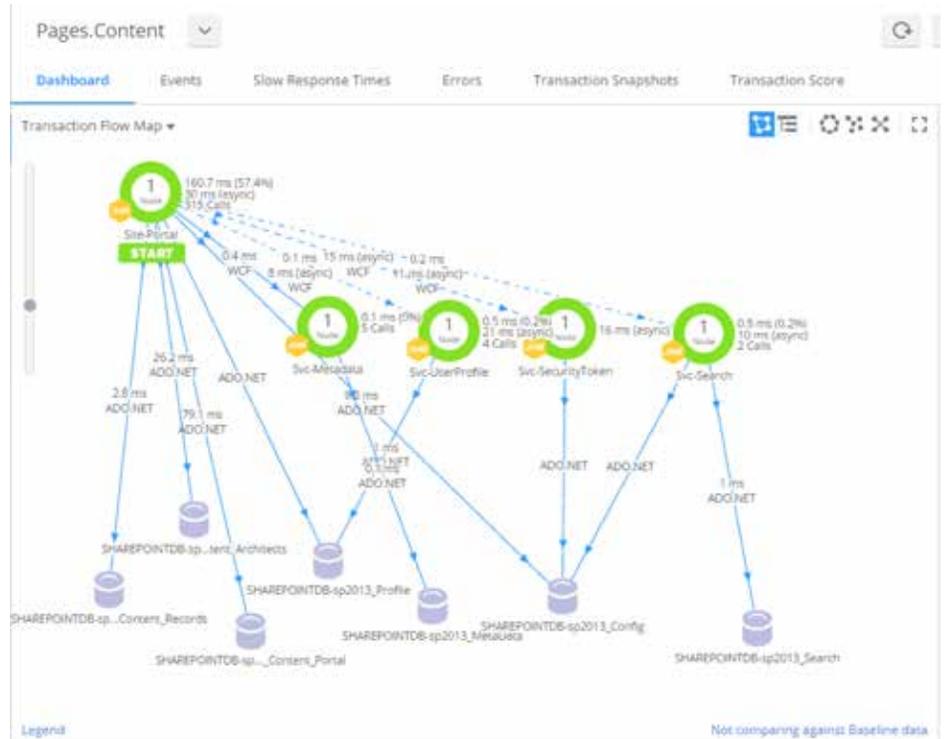
This works alongside the EUM Dashboard. The rules group activity is in buckets, similar to APM transactions.

Pages.Content

This highly complex flowmap shows how Metadata, UserProfile, Search and SecurityToken services must be there and working to display a single content page. This flowmap also shows a Business Transaction where some pages leverage Managed Metadata Service to render columns from Term Store, some leverage User Profile Service to render information about users, and some leverage Search Service to query content from the index.

Pages & AJAX Requests

Type	Name	Requests	Requests per Minute	End User Response Time (ms)	DOM Ready Time (ms)	First Byte Time (ms)	Page views with JavaScript Errors
ClientOM Query		1,013	5	476			284
ClientOM REST Query		131	2	371			156
Pages.Application		84	1	1,280	1,188		374
Pages.Application		4,887	20	101			390
Pages.Application		274	7	1,082	888		344
Pages.Content		4,148	17	3,256	1,881		251
Pages.Content		48	2	881			819
Pages.Content		127	1	2,821	1,772		685
Pages.OTCServices		181	6	302			294
Pages.ListView		4	4	178			188
Pages.Search		65	1	4,762	1,230		480
Pages.Upload		28	1	862	834		371
Pages.Upload		10	1	1,624			1,207
Pages.WebMembers		1,478	7	251			289
diagnostics.aspx		389	2	453			664



Pages.OfficeOnlineDocumentRender

This details the interactions between OWA and SharePoint to render a document.

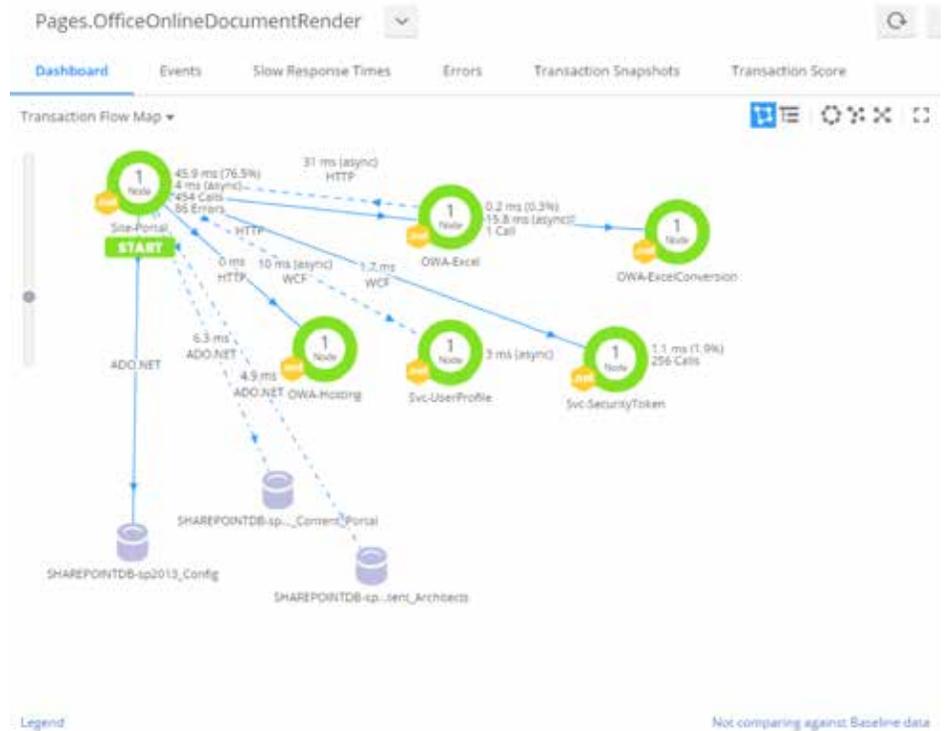
In SharePoint 2013 and 2016 OWA, rendering works this way:

The user clicks on the document in the UI:

```
2016-10-06 01:56:34 127.0.0.1 GET /_layouts/15/WopiFrame.aspx
sourcedoc=/Documents/AppDynamics%20Configuration%20Comparer.docx&action=default 443 0#.w|danielo\sp_setup 127.0.0.1Mozilla/4.0+(-compatible;+MSIE+7.0;+Windows+NT+6.2;+WOW64;+Trident/6.0;+Touch;+.NET4.0E;+.NET4.0C;+.NET+CLR+3.5.30729;+.NET+CLR+2.0.50727;+.NET+CLR+3.0.30729) https://sp2013appd-portal.danielo.com/Documents/Forms/AllItems.aspx 200 0 0 109
```

- The user is redirected to a wrapper page that embeds a full-screen iframe that contains a redirect to URL on the OWA server.
- The OWA server application receives pointer to the document and retrieves the document from the SharePoint website.

This flowmap shows that the OWA rendering page uses calls to the OWA-Hosting component to find out where OWA is, and then leverages Excel and ExcelConversion components to render an Excel document. SecurityToken service is involved in authenticating the request on behalf of the invoking user.



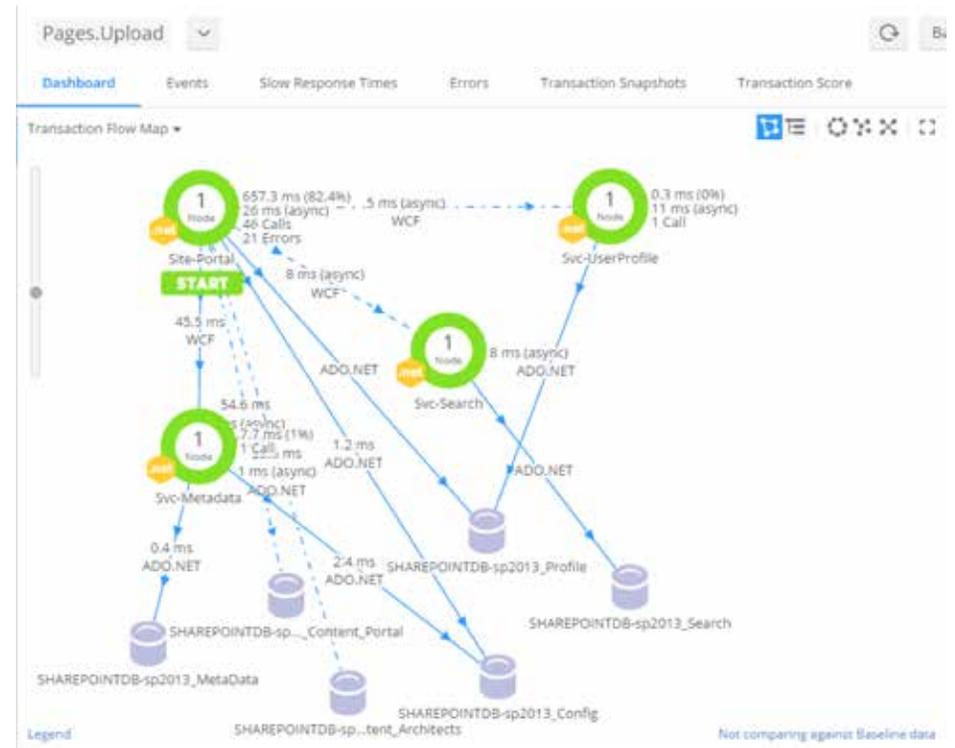
Pages.Upload

This deceptively simple display function is built on one of the most complex flowmaps.

SharePoint file upload can be done via:

- Single file upload in a page or page dialog
- Multiple file upload in a page or page dialog
- HTML5 drag and drop page upload into a drop area on a document library
- SharePoint Designer via Web Services
- Programmatically via Client Object Model
- Windows Explorer via FrontPage RPC

The upload flowmap shows that there is considerably more to upload than just bytes to the server. User Profile service is used to get information about the uploading user. Search service is used to look for the document elsewhere or even in the current destination. Metadata service is used to fill in terms from Term Sets if the document library uses them.



WCF.ClientOM.Query

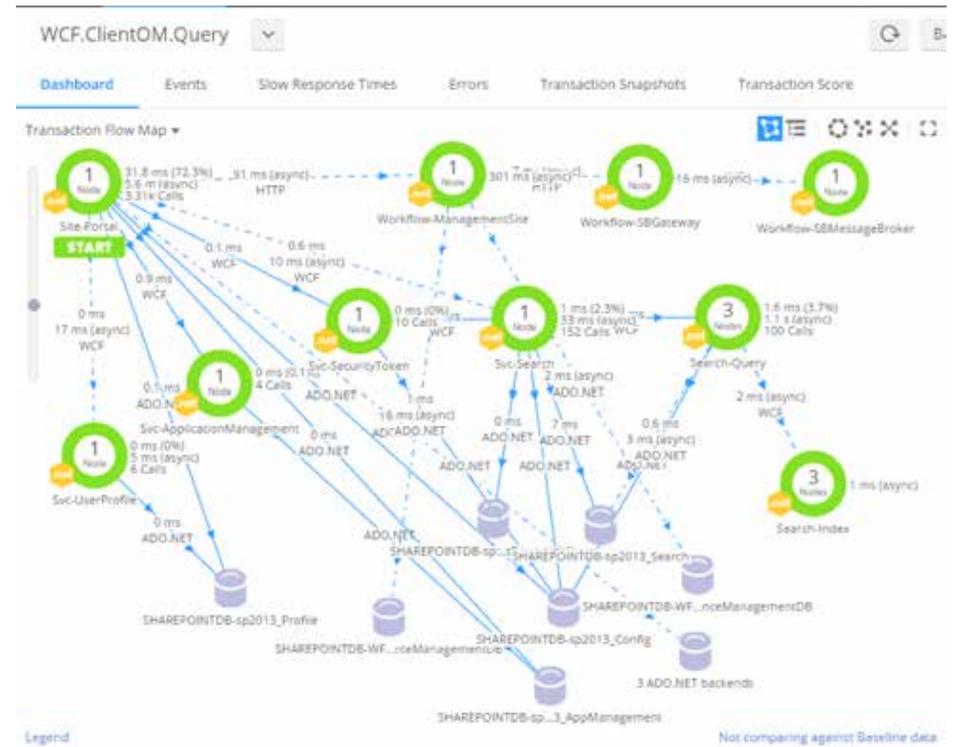
This flowmap shows the endpoint to which all calls via Client-side Object Model (CSOM) come to. It is hosted at C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\15\isapi\Client.svc. The service only has a couple of big methods, this being one of them. Input is a stream filled with XML.

This gets invoked a lot from the UI, including most importantly during file upload, search requests and workflow starts.

There are a couple of data collectors that are configured which try to extract the type of the call, but since the payload is XML and contents change widely, it is difficult.

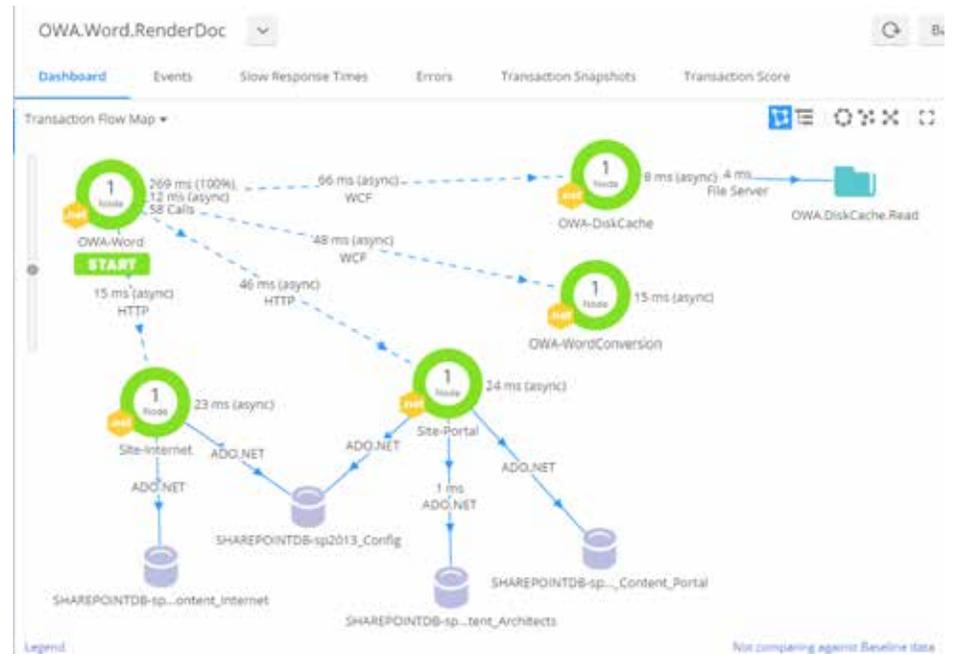
Here is the reflection of how many different things can get called via CSOM:

- Calls to User Profile to look up user
- Calls to SharePoint 2013 Workflow Manager to start workflows
- Calls to Search to issue search requests
- Calls to the site itself to upload file



OWA.Word.RenderDoc (Word)

OWA Word document rendering is started by a web page called via Iframe from Content Web Application. It turns around and talks to Content Web Application to get the document, uses WordConversion to convert it, and caches it with DiskCache service, which puts it into the file system.



Daniel Odievich has been involved in programming computers since age 7, loading software into Soviet-made computers from tape decks and modifying binaries to get around game copy protection. Despite the interest in computers, he also learned how to sew backpacks and down parkas, weld pipes, install roofing, lay bricks, pour concrete floors and seal foundations with hot asphalt.

Since mid-90ies, he worked as software engineer and professional services consultant, focusing on custom applications and software integration projects. At Microsoft, he spent a lot of time designing, building and debugging various systems for governments and large enterprises. While working on real world problems, he learned how Windows, SQL Server, SharePoint and .NET Framework work and wrote a couple of MSDN whitepapers on the topics of web development and SharePoint Workflows

Since joining AppDynamics Professional Services, he can be found helping Fortune 500 customers get the most out of their applications and helping his fellow consultants with challenging problems.



A Quick Look Back

In chapter 1, we dug into the history of SharePoint and business transactions.

In chapter 2, we explored the functions of APM and EUM in helping operations teams monitor SharePoint using AppDynamics.

In chapter 3, we tore SharePoint down to its components to look at each in detail.

In chapter 4, we saw how to troubleshoot the eight most common problems related to those components.

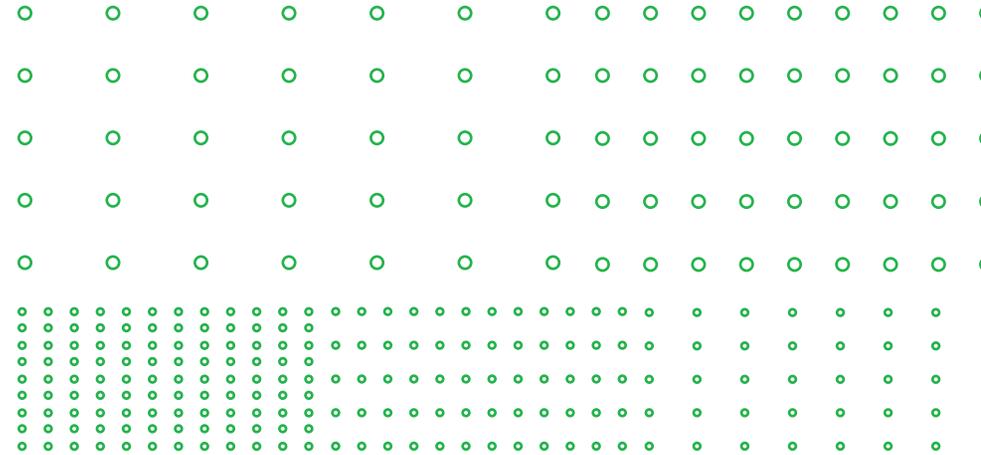
Finally, in this chapter, we examined how to set up the AppDynamics APM coverage and became familiar with resulting transactions and their flow maps.

After the Configuration

The AppDynamics platform is the most comprehensive solution for monitoring and managing all of the .NET applications connected to your SharePoint server. As you become more familiar with its capabilities, you will be able to instantly call up real-time performance visibility and pinpoint the root cause of code bottlenecks in seconds.

Contact the experts at AppDynamics to unlock the full range of capabilities at your command. You want to dig deeper into the components of the OWA, or you might just want to keep the front-end sites and services looking good. Whatever your SharePoint goals are, the team at AppDynamics will help you get there.

START A FREE TRIAL AT
APPDYNAMICS.COM



About AppDynamics

AppDynamics is the Application Intelligence company. With AppDynamics, enterprises have real-time insights into application performance, user performance and business performance so they can move faster in an increasingly sophisticated, software-driven world. AppDynamics' integrated suite of applications is built on its innovative, enterprise-grade App iQ Platform that enables its customers to make faster decisions that enhance customer engagement and improve operational and business performance. AppDynamics is uniquely positioned to enable enterprises to accelerate their digital transformations by actively monitoring, analyzing, and optimizing complex application environments at scale which has led to proven success and trust with the Global 2000. For more information, visit appdynamics.com.

APPDYNAMICS

appdynamics.com