HOW TO DO A **PERFORMANCE AUDIT** OF YOUR **.NET WEBSITE**

Managing website performance can be a complex undertaking: there are many factors that can play a role, from server resources to application code to network bandwidth. When tasked with troubleshooting an underperforming .NET website, you may not know where to start.

Improving website performance can be straightforward once you identify the root cause. In this eBook, we'll detail how to implement a performance audit of your .NET website. By using our methods, you can identify the root cause of your performance issues.

Outline

Our website performance audit is based on these five steps:

1. **Understand the problem**
2. **Analyze specific page performance**
3. **Analyze server side performance**
4. **Analyze browser resource performance**
5. **Check for site-wide performance issues**

2

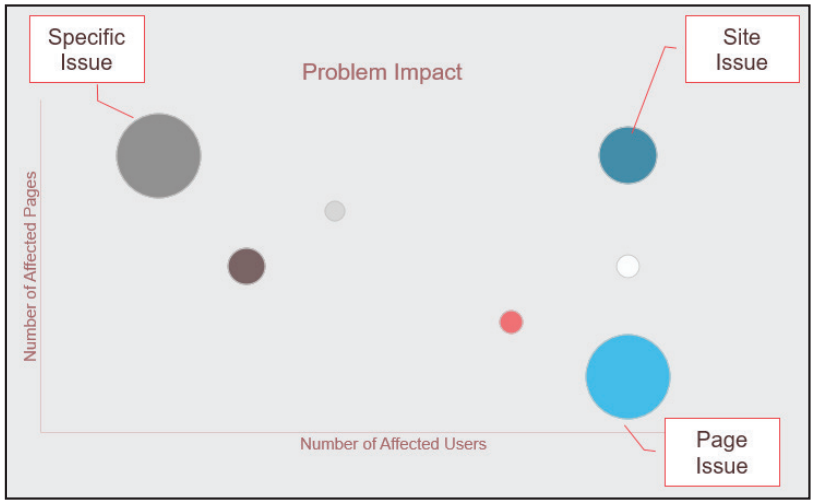
Understand the problem

It's difficult to fix a problem when you don't know the cause.

As the manager of a website, you'll often receive vague messages and mixed signals, such as:

- "The website is really slow today."
- "The website runs fine, but when I login, it becomes really slow."
- "Every afternoon, the site just hangs."

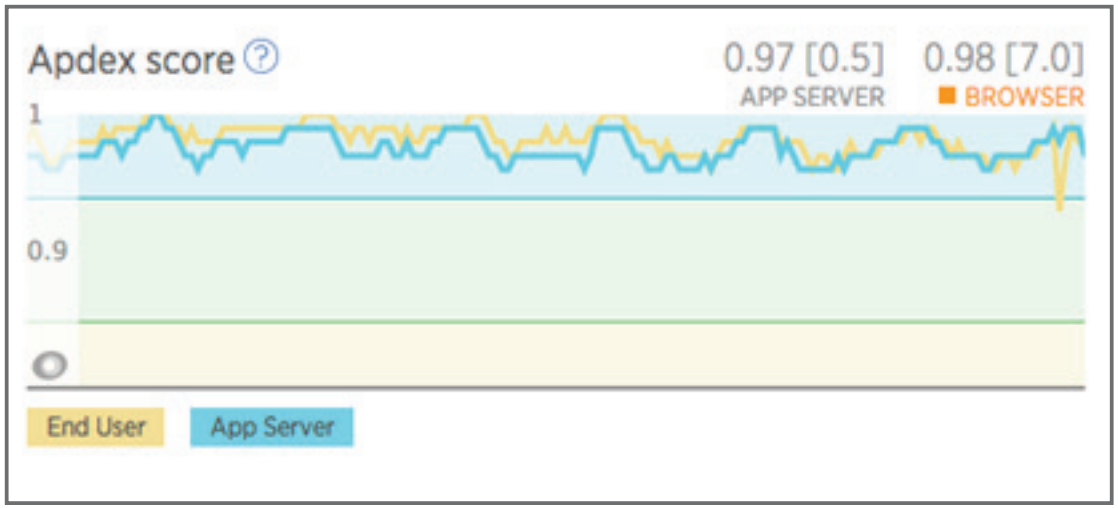
Refer to this chart to classify your problem:



The problem types are:

1. **Specific issue:** high number of affected pages, but low number of affected users
2. **Page issue:** low number of affected pages, but high number of affected users
3. **Site issue:** high number of affected pages, high number of affected users

In this eBook, we'll detail tools you can use for each step of the performance audit.



NEW RELIC

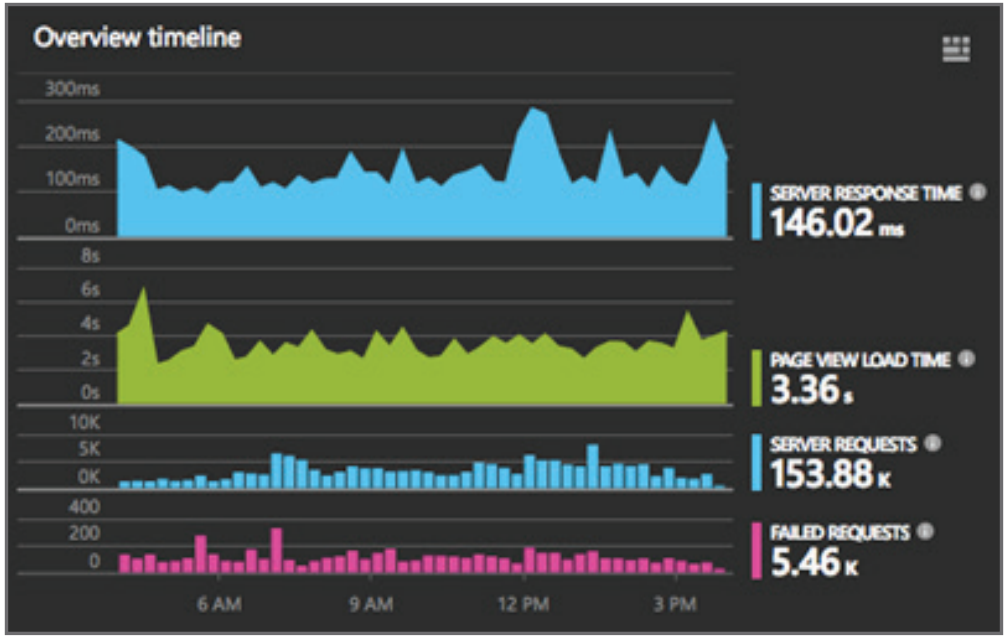
newrelic.com

How it Works: Add an agent to your server. The agents insert "beacons" into pages, which track performance metrics when pages are served from your site.

Cost: Free, with paid plans

How to Use: Review the New Relic "Apdex" score for key pages. This score shows you the ratio of satisfied to dissatisfied visitors. The score ranges from 0 to 1, where 1 is a perfect score.

Compare Apdex scores across pages. This helps you determine whether you have site-wide or page-specific issues. For instance, a high Apdex score on the homepage, paired with a low Apdex score on a product page means that further investigation is required on the latter.



VISUAL STUDIO APPLICATION INSIGHTS

azure.microsoft.com/en-us/services/application-insights

How it Works: Add an agent to your .NET application. The agent injects code into HTTP requests to collect performance data. It also ingests your server's application logs. Custom reports are provided in Azure Portal.

Cost: Free, with paid plans.

How to Use: Review metrics such as average server response time and average page load time to understand whether you have a site-wide issue.

5

Analyze specific page performance

After you isolate your performance issue, you'll have a set of specific pages to analyze further. The keys to understanding page performance are:

- Recording data over time to establish a performance baseline
- Breaking down page load time into its component parts, to identify performance culprits

CHROME LOAD TIMER

github.com/alex-vv/chrome-load-timer

How it Works: A Google Chrome browser extension that measures page load time and displays it in the toolbar.

Each time you visit a page, the load time statistics appear once the page has finished loading.

Cost: Free.

How to Use: Visit key pages multiple times (e.g. 5-20+ times) and record the load times in a spreadsheet. Cross-check load times against other pages, to determine whether you have page-specific vs. site-wide issues. Look for unusual issues, such as delays in DNS (domain name system) requests.

Load timings (ms)

Event	When	How long	Sum
Redirect	0	0	0
DNS	107	0	0
Connect	107	307	307
Request	414	3326	3633
Response	3740	270	3903
DOM	3745	4295	8198
Interactive	6705	0	-
Content loaded	6705	28	-
Load event	8040	12	8210

The grades from WebPagetest indicate that this page has issues to address.

WEBPAGETEST

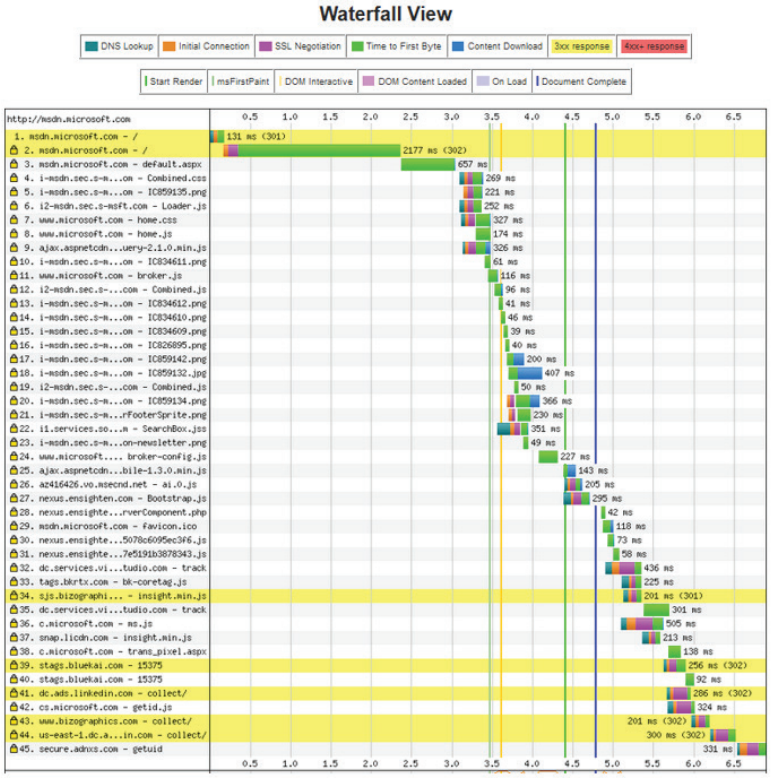
webpagetest.org

How it Works: A free service, hosted at WebPagetest.org.

Enter your URL, choose connection type and add in custom headers if desired. WebPagetest provides scores for numerous aspects of page load time.

Cost: Free.

How to Use: Record data points in a spreadsheet. Refer to WebPagetest's "Waterfall View" to understand where performance issues are occurring. For instance, the downloading of a large image may take the majority of total page load time.

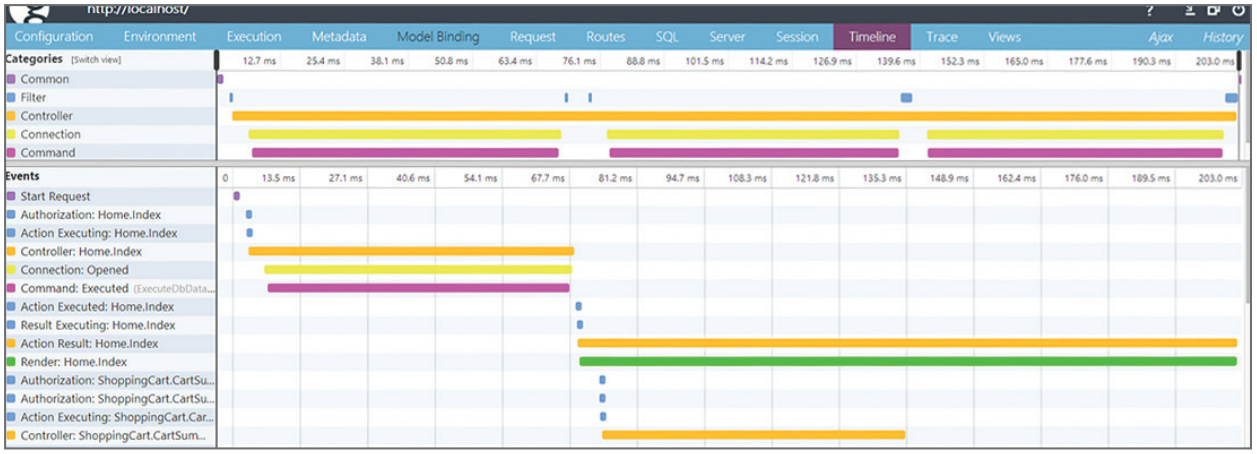


The Waterfall view can highlight the root cause of a page's performance issues.

Analyze server side performance

For analyzing server side performance, the following questions need to be answered:

- Where is time being spent on ASP.NET pages?
- How long is it taking for API calls to complete?
- How many database queries are being generated and how long are they taking?
- How much server CPU and memory is being consumed?



GLIMPSE

getglimpse.com

How it Works: A free and open source .NET tool that is installed as a component on your server. It runs as a diagnostics tool against your server code.

Cost: Free.

How to Use: Add logging calls around suspect code and identify the volume and completion time of database queries.

Further Details: Glimpse

To debug server code, add System.Diagnostics.Trace.WriteLine() calls around suspect code. These calls produce valuable debugging output that can surface performance culprits (e.g. code that is taking too long to complete).

For DNN websites, a DNN-specific version of Glimpse is available, called dnnGlimpse. As seen below, you can insert GetLogger() calls around suspect code:

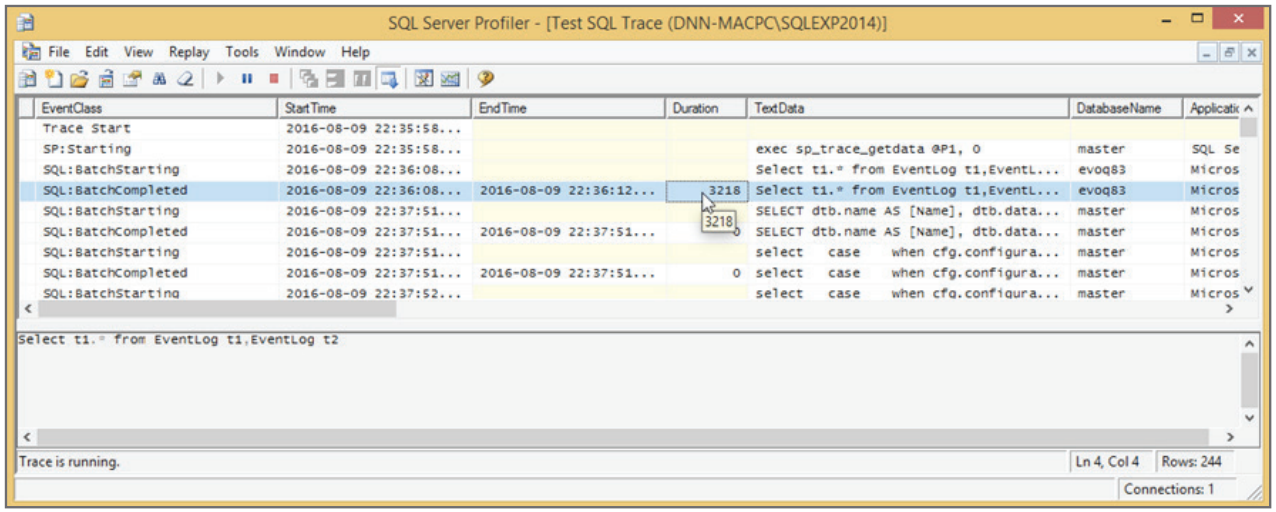
_View.cshtml DesktopModules/MyBadDeveloper/MySlowModule

```
@using DotNetNuke.Common;
@*Copyright (c) 2016 by MyBadDeveloper *@
<div class="dnnForm dnnEdit dnnClear" id="dnnEdit">
    <fieldset>
        <div class="dnnFormItem">
            <div>My Slow Module</div>
        </div>
    </fieldset>
@{
    DotNetNuke.Instrumentation.DnnLogger.GetLogger("DNN.Trace").Debug("Start Loading MySlowModule Razor Script");
    <div>in c# razor script - doing slow stuff</div>
    System.Threading.Thread.Sleep(4000);
    DotNetNuke.Instrumentation.DnnLogger.GetLogger("DNN.Trace").Debug("End Loading MySlowModule Razor Script");
}
```

In the Glimpse trace output, we can confirm that the Razor script is the culprit:

In addition to analyzing application traces, Glimpse is useful in spotting problematic database queries. In this instance, Glimpse identifies a database query that is taking too long to complete:

To download DNN Glimpse:
https://dnnglimpse.codeplex.com/



SQL SERVER PROFILER

msdn.microsoft.com/en-us/library/ms181091.aspx

How it Works: SQL Server Profiler ships with SQL Server. It creates database traces and analyzes trace results. Events are saved in a trace file that can later be analyzed or used to replay a specific series of steps when trying to diagnose a problem.

Cost: Free, with your purchase of SQL Server.

How to Use: Works well when using recognizable naming standards for stored procedures. Carefully set your filters; otherwise, you'll have to sift through a sea of information.

Analyze browser resource performance

While server side performance considers resource contention on your web servers, browser resource performance refers to the processing performed by visitors' browsers. As website managers, we're responsible for browser resource issues when we ask browsers to do too much work.

Using two of the aforementioned tools:

- Chrome Load Timer
- WebPagetest

Run a series of tests on your suspect pages. Look for oversized or broken resources:

- Large image files
- Large CSS files
- HTTP 404 errors ("Not Found")
- HTTP 401 errors ("Unauthorized")

Unless you're running a photography site for professional photographers, your images don't need to be 800 KB in size. An image of this size can take a long time to download, especially for visitors on slower connections.

When measuring your page via WebPagetest, be sure to review:

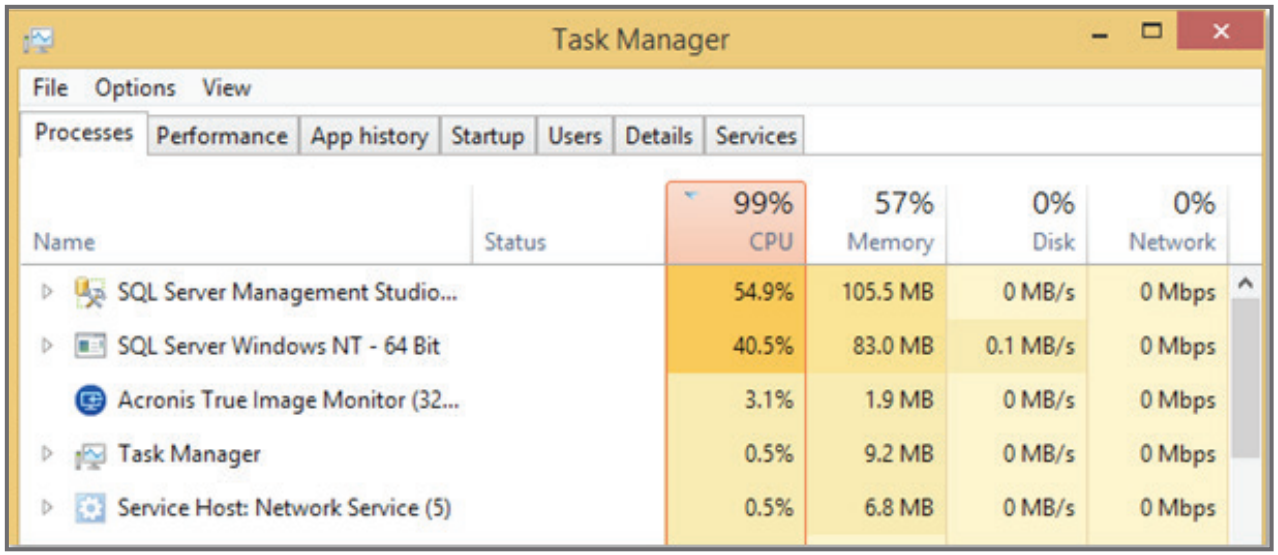
- **First Byte:** The time it takes the browser to receive the first byte of HTML from the server
- **Start Render:** The time it takes the browser to receive enough data to begin rendering the page
- **Document Complete:** The time it takes for all resources to load
- **Fully Loaded:** The time it takes for the page to fully render

For all issues you uncover, note them in a document and determine whom you need to follow up with (e.g. Marketing, Sales, etc.).

13

Check for site-wide performance issues

If your server is consuming 99% CPU running a long database query, chances are all pages on your site will be slow to load. In this final step, audit server-level resources to determine whether any hardware or software issues are affecting your site at a server-wide level.

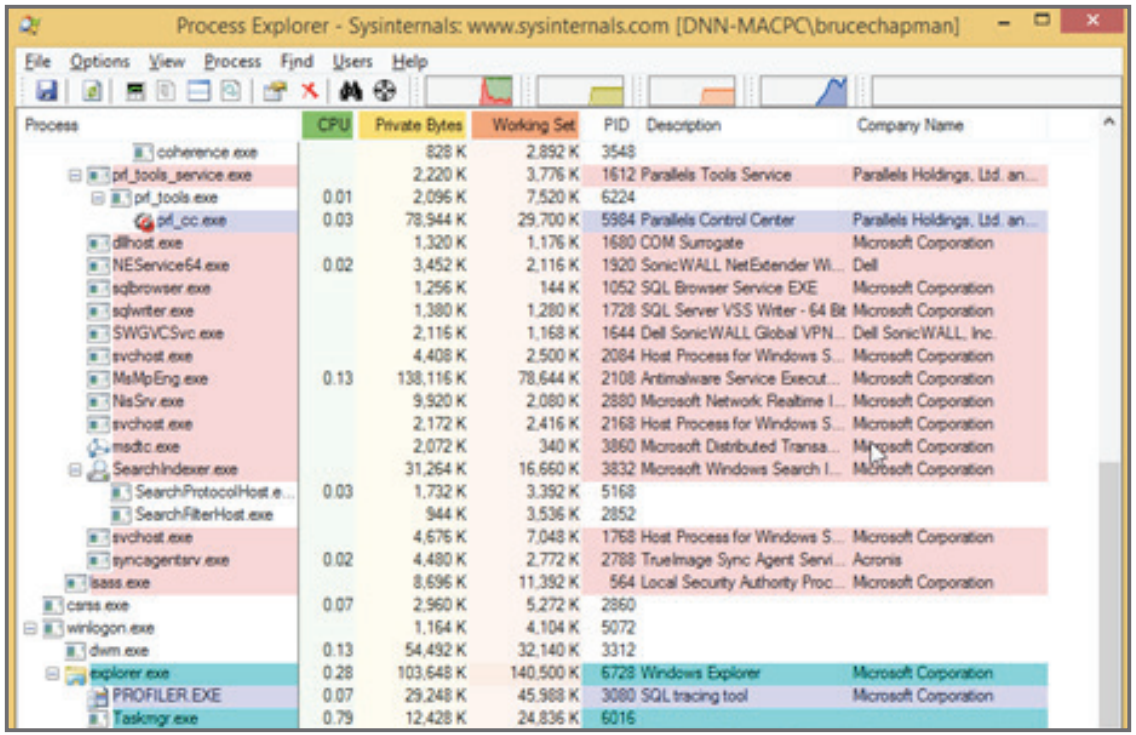


WINDOWS TASK MANAGER

How it Works: A utility that details all of the running processes on a Windows host. In the example above, 95.4% of the CPU is consumed by SQL Server, which indicates that a long-running query may be running.

Cost: Free.

How to Use: For more information, see "How to use and troubleshoot issues with Windows Task Manager."



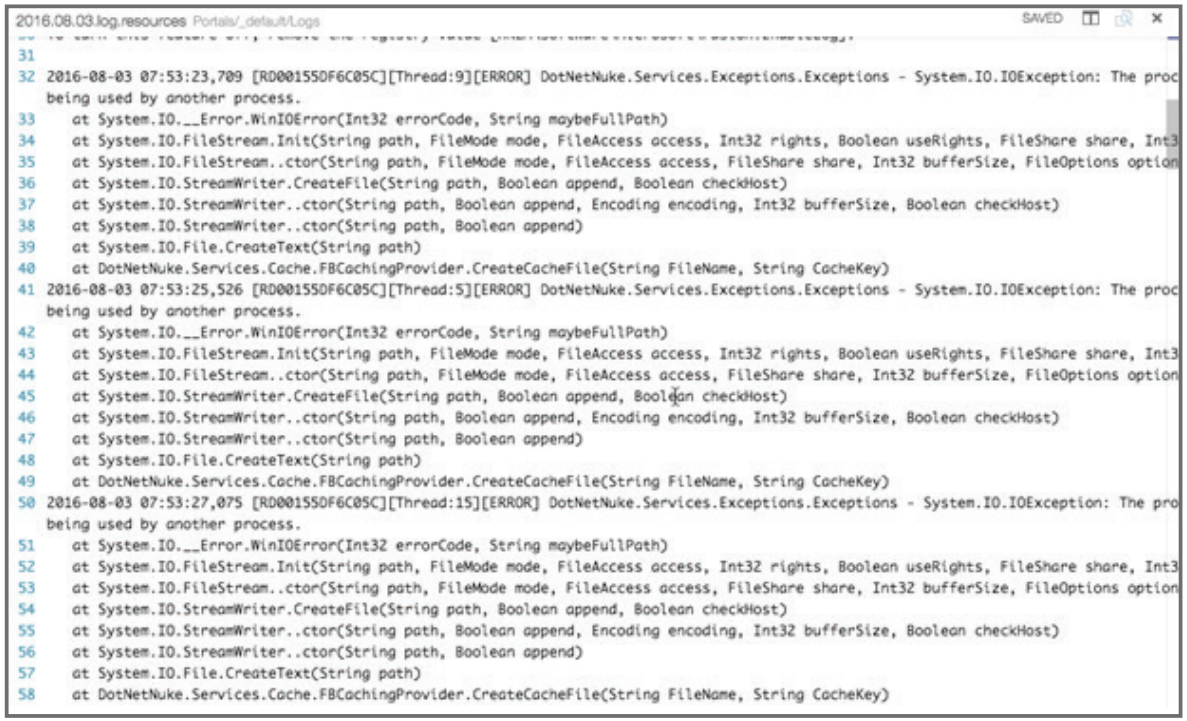
PROCESS EXPLORER

technet.microsoft.com/en-us/processexplorer

How it Works: Free download for Windows servers. Think of Process Explorer as an advanced version of the Windows Task Manager. It will show you the individual threads running on the server.

Cost: Free.

How to Use: For administrators or developers who can decipher system calls, use Process Explorer to inspect individual threads. You can identify long-running calls spot possible memory leaks.



LOG4NET

How it Works: An exception logger that's built-in to DNN.

Cost: Free.

How to Use: Use to spot problematic or commonly occurring exceptions. While exceptions are not visible to end users, they could impact the performance of your site. In the image below, the server has a locking condition, which is reflected in the exception about a resource "being used by another process."

The Audit Report: Bringing it all together

During each step in our five-step process, you'll want to document your findings. At the conclusion of the audit, you'll have an audit report, which can be shared with other groups in your organization. In addition to the audit findings, a "Conclusion and Next Steps" section should highlight the key problem areas, along with detailed steps on how to address them.

Performance Audit — Date 11-Aug-16

Site: http://example.com

Page Speed Tests

Test	Page	Total	Test	Page	Total
	1 Home	11.3		1 Products	2.1
	2 Home	6.56		2 Products	2.2
	3 Home	7.98		3 Products	2
	4 Home	8.34		4 Products	1.7
	5 Home	9.56		5 Products	1.6
	Avg	8.748		Avg	1.92
	Median	8.34		Median	2

Page Inspection

	Home		
	First Request	Second Request	Diff
Time to first Byte	4.922	4.257	-0.665
First Render	6.49	4.796	-1.694
Doc Complete	10.045	5.572	-4.473
Loaded	10.17	5.752	-4.418
Requests	37	3	-34
Size (KB)	2544	17	-2527

Slow Queries

Page	Time	SP or SQL
All	3.2	select t1* from EventLog t1. EventLog t2
Home	2.1	BEGIN WAITFOR DELAY '0:00:02' select top 1 * from users:

Oversized Resources

Page	Time	Size (kb)	Resource
Home	1.2	859	Chrysantheum.jpg
Home	3.6	234	header-shadow.jpg
Home	1.7	85	cavalier.tv.jpg

Errors and Exceptions

Time	Error

The Evoq CMS from DNN

At DNN Software, we've been working on .NET websites for over 10 years. Evoq is a leading .NET CMS that customers use to manage thousands of websites, intranets and portals. Evoq provides a number of features for website performance, including page caching, module caching and web farms.

In addition, Evoq OnDemand is a fully-managed, cloud-based CMS powered by our close partnership with Microsoft Azure. For more information about DNN, visit http://www.dnnsoftware.com or call 650.288.3150. Thanks for reading.