

Sitecore CMS 7.0 or later CMS Diagnostics Guide

A developer's guide to diagnosis of Sitecore CMS performance



Table of Contents

Chapter 1	Introduction	4
Chapter 2	Diagnostic Procedures	5
2.1 E	Browser Page Test	6
2.1.1	Setup	6
2.1.2	How to Use the AOL Pagetest Browser Plug-in	6
Cap	turing Information	7
First	t Pass — Cleared Browser Cache	7
Sec	ond Pass — Refresh Browser Page	10
Thir	d Pass — Recently Visited Webpages	11
2.1.3	Interpreting the Results	12
Wat	erfall Tab	12
Che	cklist Tab	13
Info	rmation about Solving Failures and Warnings	15
Opti	mization Report Tab	16
Load	d Details Tab	16
2.2 li	nvestigating Page Performance — Using the IIS Logs	17
2.2.1	Required Skills	17
2.2.2	Symptoms	17
2.2.3	Talk to the Partner / Customer	17
2.2.4	Procedure to Query IIS Logs for Long Running Requests	17
2.2.5	Understanding the Results	18
2.2.6	Sitecore Recommendation	19
Rep	ort Findings	20
2.3 F	Rendering Performance	21
2.3.1	Required Skills	21
2.3.2	Symptoms	21
2.3.3	Procedure to Use the Sitecore Stats Page	21
2.3.4	Understanding the Results	22
2.3.5	Sitecore Recommendation	22
Rep	ort Findings	23
2.4 li	nvestigating Sitecore Memory Usage	24
2.4.1	Required Skills	24
2.4.2	Symptoms	24
2.4.3	Parsing the Sitecore Log(s) using Sitecore Log Analyzer	25
Gett	ting the Values of Process\Private Bytes and # Bytes in all Heaps Counters	25
Gra	phing the Results	26
2.4.4	Understanding the Results (Graph)	29
2.4.5	Notes	34
2.5 S	Sitecore Pipeline Profiling	35
2.5.1	Required Skills	35
2.5.2	Procedure to Use the Sitecore Pipeline Profiling Page	35
2.5.3	Overview	35
2.5.4	Usage	36
2.5.5	Understanding the Results	36
2.6 5	Sitecore Debugger	38
2.6.1	Procedure to activate Sitecore Debugger	38
2.6.2	Usage	39
Chapter 3	Search Diagnostics	43
3.1 F	FillDB	44
3.1.1	Generating Items Using the FillDB Page	44



3.2	Verbose Logging	46
3.3	LingScratchPad	47
3.3.1	Required Skills	47
3.3.2	Usage	47

The information contained in this document represents the current view of Sitecore Corporation on the issues discussed as of the date of publication and is subject to change at any time without notice. This document and its contents are provided AS IS without warranty of any kind, and should not be interpreted as an offer or commitment on the part of Sitecore, and Sitecore cannot guarantee the accuracy of any information presented. SITECORE MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

The descriptions of other companies' products in this document, if any, are provided only as a convenience to you. Any such references should not be considered an endorsement or support by Sitecore. Sitecore cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers.

All trademarks are the property of their respective companies

©2013 Sitecore Corporation. All rights reserved.



Chapter 1

Introduction

This guide is designed as a companion guide to the Sitecore CMS Performance Guide. It provides you with a series of diagnostic procedures that can help you identify performance issues, as well as a means to measure performance gains through tuning.

This manual contains the following chapters:

- Chapter 1 Introduction
- Chapter 2 Diagnostic Procedures
- Chapter 3 Search Diagnostics



Chapter 2

Diagnostic Procedures

Diagnostic procedures are a series of tests that help identify performance issues in a Sitecore implementation.

The procedures defined are complementary to the CMS tuning. By running the diagnostic procedures before and after tuning the Sitecore CMS, performance improvements can be recorded.

This chapter contains the following sections:

- Browser Page Test
- Investigating Page Performance Using the IIS Logs
- Rendering Performance
- Investigating Sitecore Memory Usage
- Sitecore Pipeline Profiling



2.1 Browser Page Test

"The size of the average web page of the top 500 websites has more than quintupled since 2003. From 2003 to 2009 the average web page grew from 93.7K to over 507K (see Figure 1), over 5.4 times larger (Domenech et al. 2007, Flinn & Betcher 2008, Charzinsk 2010). During the same six-year period, the number of objects in the average web page more than doubled from 25.7 to 64.7 objects per page. Longer term statistics show that since 1995 the size of the average web page has increased by 35 times, and the number of objects per page has grown by 28 times." - http://www.websiteoptimization.com/speed/tweak/average-web-page/

What the previous statement means, is that in today's webpages it is crucial to understand the number and size of objects that are loaded into a webpage. Also, how a web server is setup to cache objects and minimize the number of requests that are made is important.

There is a lot of information that can be gathered by analyzing what takes place when a webpage is loaded into a browser. Information about what is being requested, time to response, size of objects, and so on is available. Also, how the server is setup in terms of caching, compression, CDN usage, and keep-alive information can be viewed.

This task looks at using an open source plug-in, AOL Pagetest, along with how to interpret the results.

2.1.1 Setup

The AOL Pagetest browser plug-in can be downloaded from: <u>http://sourceforge.net/projects/pagetest/files/</u>

Once it has been downloaded, install it in its default location. The AOL Pagetest plug-in appears in the Tools menu and works with IE 7 and later.

Note

If IE was opened during the installation, you must restart IE for the AOL Pagetest to appear in the Tools menu.

2.1.2 How to Use the AOL Pagetest Browser Plug-in

To launch AOL Pagetest:

1. Launch IE.



2. In the **Tools** menu, *click* **AOL Pagetest**.

😒 AOL Pagetest	
File View Tools Help	
waterraii	(
	H I

Capturing Information

There are 3 useful page loading scenarios that are gathered during this exercise to see how the webpage reacts to a cleared browser cache, a webpage reload, and a webpage request with a loaded cache.

• Cleared browser cache — this simulates the first time a page is visited from the browser being used.

To clear the browser cache for **IE 8** and later:

- 1) In the Safety menu, click Delete browsing history...
- 2) Clear Preserve Favorites website data, and select Temporary Internet Files, Cookies, and History.

To clear the browser cache for **IE 7**:

- 1) In the **Tools** menu, click **Internet Options**.
- 2) Under Browsing History, click Delete...
- 3) To delete your cache, click **Delete files**...
- 4) Click Close, OK.
- Webpage reload or refresh

This forces requests to be made to objects in cache, resulting in 304 status codes. When a 304 status code appears, a request is still made for an object, even though no download occurs.

• Accessing a recently visited webpage — typing the URL

This results in a page with a much reduced request chain, since items that were previously loaded into the browser cache are accessed without making any requests.

First Pass — Cleared Browser Cache

- 1. Launch IE.
- 2. Clear the browser cache.
- 3. Launch AOL Pagetest.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.



- 4. Type in the URL you wish to test for example: http://www.sitecore.net.
- 5. The AOL Pagetest window contains 4 tabs Waterfall, Checklist, Optimization Report, and Load Details.



Observations:

- The Waterfall tab shows a full request made to each object in the webpage.
- That assets such as CSS and JS files have not been combined, so multiple requests are being made to download them. By combining all of your CSS and JS files, the time line is pushed to the



left, due to a single request being made, shortening the time it takes to render the page.

Waterfal	Checklist	Optimization	Report	Load Details								
			Cache SI	tatidUse a CDN	mbine CSS/	GZIP text	mpress Imag	Keep-Alive	Cookies	Minify JS	No ET ags	
🕀 http:	//www.siteco	ore.net/	77%	0%	60%	100%	78%	100%	1%	76%	6%	
E www	v.sitecore.net	-7				✓			 Image: Image: Ima		✓	
E www	v.sitecore.net	- screen.css	- 🔥	8	8	✓		✓	- 8		8	
E www	v.sitecore.net	 forms_v3.cs 	<u>∧</u>	8	8				- 😡		8	
± www	v.sitecore.net	 jquery.fancy 	· 🔥	8	8				- 8		8	
± ₩₩₩	v.sitecore.net	 ie_common. 		8	• 🕄	✓			• 😣 -		8	
± www	v.sitecore.net	 visitor.css.a: 	9									
E www	v.sitecore.net	- Default.css		- Q	• 😣				- 63		8	
± www	v.sitecore.net	- Default.css	- 🔥	8	8				- 8		8	
± ₩₩₩	v.sitecore.net	 Custom.css 	<u> </u>	- Q	8				- 63		8	
± www	v.sitecore.net	 WebResour 		8	8				- 83	8		
E www	v.sitecore.net	 prototype.js 	- 🔼	- Q	• 😣	✓			- 63	8	8	
⊞ www	v.sitecore.net	 Wfm.Control 	- 🔥	- 8					- 83	8	8	
E www	/.sitecore.net	 wfmanalytic: 	 ▲ 	- Q		✓			- 63	8	8	
E www	v.sitecore.net	 validator.js 	<u> </u>	8		✓			- 89	8	8	Ξ
E www	v.sitecore.net	 HttpCombine 	 ✓ 	- Q		✓		✓	- 63	1		
E www	v.sitecore.net	 ScriptResou 	 V. 	8						8		
E www	v.sitecore.net	 ScriptResou 		- Q		✓			- 63	8		
⊞ www	v.sitecore.net	 ScriptResou 	 V 	8		✓			- 83	- 83		
E www	/.sitecore.net	 flag_belgium 		- Q					- 63		8	
E www	v.sitecore.net	 flag_china.a 		8					- 89		8	
E www	v.sitecore.net	- logo.gif	<u> </u>	- Q				✓	- 63		1	
E www	v.sitecore.net	 flag_german 		8							8	
E www	v.sitecore.net	 flag_france. 	- V.	- Q				✓	- Q		1	
± ₩₩₩	v.sitecore.net	 flag_hungar 	 ✓ 	8					- 63		8	
E www	/.sitecore.net	 fancy_close 	A 🔼	- I 😡 - I			V	✓	- 1		1	
± www	v.sitecore.net	- flag_japan.a		8				✓			8	
± www	v.sitecore.net	 flag_denmar 		- S			4	✓	- 63		1	
± www	v.sitecore.net	 fancy_nav_ 	<u> </u>	8				✓			8	
± www	v.sitecore.net	 flag_netherlage 		- Q				✓	- 63		1	
E www	v.sitecore.net	 flag_norway 							- 69		1	
E www	/.sitecore.net	- fancy_nav_		- S			✓	✓	1		1	

Waterfall Checklist Optimization Report Load Details
Optimization Report : 04/11/11 at 11:11:25
Results for "http://www.sitecore.net/":
Page load time: 5157 seconds Time to fise Page Downloaded: 0.846 seconds Time to Start Render: 1.333 seconds Time to Start Render: 1.333 seconds Time to Document Complete: 5157 seconds Time to Fully Loaded: 5157 seconds Bytes sent out 62.812 KB Bytes received: 381.117 KB DNS Lockurs: 1
Connections: 2 Requests: 106 Redirects: 0 Not Modified: 0 Not Found: 0 Other: 0
Enable browser caching of static assets: WARNING (7.0 days) - http://www.sitecore.net/css/fancybox/jquery.fancybox-1.3.1.css?version=11.4.1.19924 WARNING (7.0 days) - http://www.sitecore.net/css/fancybox/jquery.fancybox-1.3.1.css?version=11.4.1.19924 WARNING (7.0 days) - http://www.sitecore.net/css/re_common.css?version=11.4.1.19924 WARNING (7.0 days) - http://www.sitecore.net/css/re_common.css?version=11.4.1.19924 WARNING (7.0 days) - http://www.sitecore.net/css/re_common.css?version=11.4.1.19924 WARNING (7.0 days) - http://www.sitecore.net/images/_interface/frontpage/sidebar_border.gif WARNING (7.0 days) - http://www.sitecore.net/images/_interface/frontpage/sidebar_norearrow.gif WARNING (7.0 days) - http://www.sitecore.net/images/_interface/rontpage/sidebar_border.gif WARNING (7.0 days) - http://www.sitecore.net/images/_interface/navbg.gif WARNING (7.0 days) - http://www.sitecore.net/images/_interface/navdivgf.gif WARNING (7.0 days) - http://www.sitecore.net/images/_interface/navdivgf.gif WARNING (7.0 days) - http://www.sitecore.net/images/_nav/downarow.gif WARNING (7.0 days) - http://www.si



Waterfall Checklist Optimization Report	Load Details	
Results for "http://www.sitecore.net/":		<u> </u>
Page load time: 5.157 seconds Time to first byte: 0.640 seconds Time to Base Page Downloaded: 0.846 seco Time to Base Page Downloaded: 0.846 seconds Time to Document Complete: 5.157 seconds Bytes secrive: 0.842 KB Bytes received: 381.117 KB DNS Lookups: 1 Connections: 2 Requests: 106 Redirects: 0 Not Modified: 0 Not Modified: 0 Not Found: 0 Other: 0 Base Page Response: 200	nds	
Request details:		
Request 1: Action: GET Uth http://www.sitecore.net/ Host www.sitecore.net Result.code: 200 Transaction time: 0.845 seconds Document: 1 Socket 4 Request tayle: 0.526 seconds Document: 1 Socket 4 Request size (out): 570 Bytes Response size (in): 16036 Bytes Response Dipect Size (out): 0 Bytes Response Headers: Content: Keep-Alive Content: Keep-Alive Content: Length: 15485 Expires: -1 Date: Mon, 11 Apr 2011 15:10:45 BMT Content: Recondril: 927 Set-Cookie: SC_ANALYTICS_GLIDBAL Set-Cookie: SC_ANALYTICS_SIDBAL Set-Cookie: SC_AN	3ytes ge/[peg, application/xaml+xml, image/git, image/pipeg, ap E 7.0; Windows NT 6.1; WOW/64; Trident/4.0; SLCC2; N 2kg50x5me552kkpyej; path=/; HttpOnty CDONK==TF70BEDC8230426E95AF2DE49528EFCF; exp CDONK==TF70BEDC8230426E95AF2DE49528EFCF; exp CDONK==338B85BFC4434B148800086AB43E28CC4.1; p	oplication/x-ms-xbap, application/x-shock. IET CLR 2.0.50727; .NET CLR 3.5.30725 sires=Tue, 10-Apr-2012 15:10:45 GMT; pa Jath=/

Second Pass — Refresh Browser Page

- 1. In the AOL Pagetest File menu, click New.
- 2. Refresh the browser page click refresh page, or press F5.



3. View the results in the Waterfall tab, and compare them to the results from the first pass.

Waterfall Checklist Optimization	Report Load	Details										
	0.5	1.0 1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	*
www.sitecore.net - /		1014 ms										
😟 www.sitecore.net - screen.css		📕 119 ms (30)4)									
www.sitecore.net - forms_v3.cs		82 ms (30)4)									
www.sitecore.net - jquery.fancy		60 ms (3	04)									
www.sitecore.net - ie_common.		71 ms	(304)									
 www.sitecore.net - visitor.css.as 		7 8 ms										
www.sitecore.net - Default.css		71 m	304)									
www.sitecore.net - Default.css		66 m:	: (304)									
www.sitecore.net - Custom.css		59 m	ns (304)									
www.sitecore.net - WebResour		58 m	ns (304)									
 www.sitecore.net - prototype.js 		57 (ms (304)									
www.sitecore.net - Wfm.Control		11	7 ms (304)									
www.sitecore.net - wfmanalytics		62	ms (304)									
www.sitecore.net - validator.js		6	0 ms (304)									
www.sitecore.net - HttpCombine			328 ms									
www.sitecore.net - ScriptResou			65 ms (304)									-
www.sitecore.net - ScriptResou			119 ms (3	304)								=
www.sitecore.net - ScriptResou			118 ms	(304)								
www.sitecore.net - flag_belgium			65 ms	(304)								
www.sitecore.net - flag_china.a			126 (ns (304)								
www.sitecore.net - flag_denmar			123	ms (304)								
www.sitecore.net - logo.gif			11	8 ms (304	4)							
www.sitecore.net - flag_german			1	42 ms (3	04)							
www.sitecore.net - flag_france.			6	9 ms (304	4)							
www.sitecore.net - flag_hungary				129 ms	304)							
www.sitecore.net - flag_japan.a				122 ms	(304)							
www.sitecore.net - icon_search				120 m	s (304)							
www.sitecore.net - flag_netherlag				134 m	is (304)							
www.sitecore.net - flag_poland.				66 m	s (304)							
www.sitecore.net - flag_norway				131	ms (30-	4)						
www.sitecore.net - flag_russia.a				13	0 ms (30	14)						
www.sitecore.net - flag_sweder					235 ms	(304)						
www.sitecore.net - flag_spain.a					175 ms	(304)						_
www.sitecore.net - flag_united_					121 n	ns (304)						
www.sitecore.net - downarrow.g					122 r	ns (304)						
www.sitecore.net map.ashx					2	50 ms						
www.sitecore.net - sidebaricon_					/1 n	ns						
www.sitecore.net sidebaricon_					80	ms						
www.sitecore.net - sidebaricon_					70	94 ms						
www.sitecore.netn_sitecore					75 ms							
www.sitecore.netn%20Next			_	1	30 ms		_					
					1000							

Observations:

- The objects highlighted in yellow are 30x status codes.
- The results show that cached items are requested, but not downloaded. This is expected behavior for a page that has been refreshed.
- What can be taken from this view is that those items with a 304 status code should not be requested during the 3rd pass *Recently Visited Webpages*.

Third Pass — Recently Visited Webpages

- 1. In the AOL Pagetest File menu, click New.
- 2. In the browser select the URL and hit enter do not click the browser's refresh button.



3. View the results in the **Waterfall** tab, and compare them to the results from the first and second passes.

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1,1	1.2	1.3	1,4	1.5	1.6	-11
www.sitecore.net - /														1367	ms		
www.sitecore.net - visitor.css.as									15	2 ms							
www.sitecore.netrtner%20M0												389	ms				
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.
																	- 1

Observations:

- The request tree has been reduced significantly.
- It is known that the first two requests, / and visitor.css, are dynamic and not cached.
- The third item requires investigation why is it not coming from cache?

Upon further investigation it was found that this request is a rotating image, which changes across visits. It is OK not to be in cache.

2.1.3 Interpreting the Results

Here we dive into the **Waterfall**, **Checklist**, **Optimization Report**, and **Load Details** tabs, explaining the information that is presented.

Waterfall Tab

The **Waterfall** tab presents a view of the request chain that makes up the requested objects for a web page. The requested objects are listed down the left side, and the time is listed across the top.

The colors that make up a request / response are:

DNS Lookup
Initial Connection
Time to First Byte
Content Download
Start Render
Document Complete
3xx Result
4xx Result

- DNS Lookup The time it takes to look up the IP Address for the requested URL.
- Initial Connection The time that it takes to go from the client to the server to open up a socket.



If an initial connection is made on every request, this could indicate that HTTP keep-alive has not been enabled on the server.

- **Time to First Byte** The time that it takes from making the request to receiving the first byte of the object. If the *Time to First Byte* times are extended, this could indicate that there is a problem at the server for example, performance issues, or network issues.
- **Content Download** The time required to download an object. If the Content Download times are extended, this could indicate the presence of large objects. An inspection of the size of the object may be required. For example, images that have not been optimized or compressed could have an effect on the performance of the web page.
- Start Render The green vertical line shown on the Waterfall tab. This indicates when the user starts to see content appear in the browser. Technically this represents the time when the web page has a height and width > 0.
- **Document Complete** The time when the document complete event had fired.
- **3xx Result** Objects that are highlighted in yellow indicate that 3xx status code has occurred, such as a request or cache request.
- **4xx Result** Objects that are highlighted in red indicate that a 4xx status, or error code has occurred. These should be investigated ASAP.

Checklist Tab

The **Checklist** tab is a report card of how well your website takes advantage of performance related settings and procedures.

Topics are listed from left to right in order of importance:

- A green check mark indicates a pass.
- A yellow triangle indicates a *warning*.



• A red X indicates a "failure"

waterial checking uptimization Report Luau Deta	ils									
	Cache Static	Use a CDN	Combine CSS/JS	GZIP text	Compress Images	Keep-Alive	Cookies	Minify JS	No ET ags	A
	77%	0%	60%	100%	73%	100%	0%	76%	6%	
www.sitecore.net - /				1			<u> </u>		1	
www.sitecore.net - screen.css	1	8	8	j j		j j	O		8	
www.sitecore.net - forms v3.css v3.csss v3.csss v3.csss	Ā	Ö	- Ö	j j		j j	Ö		Ö	
IF www.sitecore.net - iguerv.fancvbox-1.3.1.css	×	. Ö	. Ö	- j		j j	Ö		. Ö	
F www.sitecore.net - je common.css	i i i	ā	l 🙇	j j		j j	. Ö		ā	
www.sitecore.net - visitor.css.aspx		-	-			j j	Ā			
www.sitecore.net - Default.css	<u>^</u>	6	6				ö		Ó.	
www.sitecore.net - Default.css	×	ă	i 🧸			<u> </u>	ŏ		ă	
www.sitecore.net - Custom.css	X	ă	a di seconda di second				ă		ă	
www.sitecore.net - WebBesource.axd	1 7	ă	a di seconda di second	- <u>'</u>			ă	6		
www.sitecore.net - prototype is	Å	ă	, A				ŏ	ŏ	à	
www.sitecore.net - Wfm Controls is		ă	1			<u> </u>	ă	ă	ă	
www.sitecore.net - wfmanalutics is	×	ă		Š.		, in the second se	ă	ă	ă	
H www.sitecore.net - validator.is		ă				ž	ă	ă	ă	-
E www.sitecore.net - HttpCombiner.ashv	2	ă				ž	ă	ă		-
E www.sitecore.net - ScriptBesource.avd		ă				ž	ă	ă	<u> </u>	
Immusitecore net - ScriptResource avd	× 1	Ă					ă	ă	, v	
Immunicecore net - ScriptResource avd	1 ×	ă 🕺		<u> </u>			ă	ă		
Immunitedeore.net - conputeredeuted.and		Ă		•			ă	w w	Å	
www.stecore.net - flag_beigiun.dshix	· ·	X				×,	×		X	
www.sitecore.net - logo aff	i i i	X			7	×,	ă		X	
With stecore net flag denmark ashy	2	X			i i i	×,	ă		X	1.11
E www.stecore.net fance, close and	1 X	X				×,	×		X	
E www.stecore.net flag.germanu.ashv		× ×			i i i	×,	×		X	
E www.sitecore.net icon_cearchfield.nif	X	X				×,	×		X	
I www.sitecore.net flog frames solu		×			X X	· · ·	×		×	
I www.sitecore.net fanou nau right nng	l X	X				×,	×		×	
I www.sitecore.net flag humgaru adhu	-	X			i i i	×,	×		×	
www.sitecore.net - nag_nungary.asnx	V	×			4	×,	×		×	
E www.sitecore.net flag_apan.asnx	· · ·	×				×,	×		8	- 11
www.sitecore.net - nag_nethenands.ashx		8			4	×,	8		8	
E www.sitecore.net flag peruga ada		8				×,	8		8	
www.sitecore.net fing_norway.asnx		×			-	×,	×		×	
E www.skecore.net flag polano.ashx		×			× 1	×,	×		×	
E www.skecore.net finag_russia.asnx		×			-	×,	×		×	
H www.sitecore.net -tancy_shadow_nw.png		8				×,	8		8	
www.sitecore.net - downarrow.gir		8				×,	×		8	
H www.sitecore.net - riag_spain.ashx		2 C			4	×,	2		2	
H www.sitecore.net - riag_sweden.ashx		2			4	V	2		2	
H www.sitecore.net fancy_shadow_w.png	4	No. 1					No. 1		No. 1	
H www.sitecore.net flag_united_kingdom.ashx		No. 1			N N	V.	Sec. 1		S	
www.sitecore.net - map.ashx	<u>↓</u>	<u> </u>			8		2		<u> 8</u>	
www.sitecore.net - sidebaricon_facebook.ashx shall be added as the sidebaric on the sidebaric o	<u>↓</u>	<u> </u>				v	2		2 2	
www.sitecore.net fancy_shadow_sw.png	4	Q				V.	S		Q	
H www.sitecore.net - sidebaricon_linkedin.ashx	L 🐥	<u>v</u>					<u> </u>		<u> </u>	

The following table describes each column in the **Checklist** tab, what objects they apply to, as well as what is checked:

Column	Objects	Description
Cache Static	Applicable Objects	Any non-html object with a mime type of "text/*", "*javascript*" or "image/*" that does not explicitly have an Expires header of 0 or -1, a cache-control header of "private", "no-store" or "no- cache" or a pragma header of "no-cache".
	What is checked	An "Expires" header is present (and is not 0 or -1) or a "cache- control: max-age" directive is present and set for an hour or greater. If the expiration is set for less than 30 days, you get a warning (only applies to max-age currently).
Use A CDN	Applicable Objects	All static non-html content (css, js and images)
	What is checked	Checked to see if it is hosted on a known CDN (CNAME mapped to a known CDN network). The known CDN's are Akamai, Amazon CloudFront, Coral Cache, Edgecast, Google, Highwinds, Internap, Limelight, Mirror Image, Panther and Yahoo.
Combine CSS/JS	Applicable Objects	All css and javascript objects.



Column	Objects	Description
	What is checked	If multiple files of the same type are served, each additional css file beyond 1 subtracts 5 percent and each Javascript file beyond the first subtracts 10 percent.
GZIP Text	Applicable Objects	All objects with a mime type of "text/*" or "*javascript*"
	What is checked	Transfer-encoding is checked to see if it is gzip. If it is not, the file is compressed and the percentage of compression is the result (so a page that can save 30% of the size of it's text by compressing would yield a 70% test result).
Compress	Applicable Objects	Any object with a mime type of "image/*".
Images	What is checked	GIF — All pass. PNG — Must be 8 bit or lower —no 24-bit PNGs pass. JPEG — Within 10% of a photoshop quality 50 pass, up to 50% larger warns and anything larger than that fails. The overall score is the percentage of image bytes that can be saved by re-compressing the images.
Keep-Alive	Applicable Objects	All objects that are from a domain that serves more than one object for the page — for example, if only a single object is served from a given domain it is not checked.
	What is checked	The response header contains a "keep-alive" directive or the same socket was used for more than one object from the given host.
Cookies	Applicable Objects	All requests.
	What is checked	Any request for a static object that sends up a cookie fails. All other requests that send up cookies warn.
Minify JS	Applicable Objects	All html, javascript and json responses.
	What is checked	Javascript is run through jsmin. If the original content was gzip encoded, the minified version is also gzipped for comparison. If > 5KB or 10% is saved, it fails. If > 1KB is saved, it warns, otherwise it passes.
No Etags	Applicable Objects	All requests
	What is checked	If the response headers include an ETag header, the request fails.

Information about Solving Failures and Warnings

- Failures in the Cache Statics column indicate that the HTTP Expires Web content header is not set, refer to: *IIS Expire Web Content Header.*
- Warnings in the Cache Statics column indicate that the HTTP Expires Web content header has a value set to less than 30 days. Refer to: *IIS Expire Web Content Header.*
- Failures in the Combine CSS /JS column indicate that CSS and or JS files are not combined into fewer files, indicating that multiple requests are being made to pull in these objects.
- Failures in the GZIP text column indicate that static compression is not enabled on the web server. Refer to: *Enable IIS Static Content Compression.*

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.



• Failures in the Keep-Alive column indicate that HTTP keep-alive is not enabled on the web server, causing a connection having to be initiated for every object requested. Refer to: *Enable IIS HTTP keep-alive*.

Optimization Report Tab

The **Optimization Report** tab provides general information related to load times for the page, as well as where improvements can be made (based on the topics in the **Checklist** tab). In some cases, such as compression settings, an estimate of how much savings in terms of size is provided.

The **Optimization Report** tab is a textual view of what is presented in both the **Waterfall** and **Checklist** tabs.

Load Details Tab

The **Load Details** tab is a detailed account of what happens with every request. This information is useful for narrowing down issues with a particular request.



2.2 Investigating Page Performance — Using the IIS Logs

This Investigating Page Performance Task is designed to find pages that take longer than 4000 ms (4 sec) round trip, using the IIS logs. The results are recorded and used as a starting point for further investigation into which rendering component, or components, may be the culprit leading to poor performance.

The advantage of parsing information from the IIS logs vs. measuring page performance directly is that you also are able to look at geo-location information (possible network issues that occur due to location), and issues that may occur during peak usage (indicators of capacity problems).

2.2.1 Required Skills

- A working knowledge of Log Parser.
- A working knowledge of the IIS Logs.

2.2.2 Symptoms

• Slow page round trip times.

2.2.3 Talk to the Partner / Customer

The first step in determining where problem / slow loading pages exist is to ask the partner / customer / website owner if they are aware of any pages that appear to be slow. Record the URIs for further investigation into what is causing the performance issues.

Also, find out what are the peak run times for the website. This information is used to determine if the website has the capacity to handle load during peak usage.

2.2.4 **Procedure to Query IIS Logs for Long Running Requests**

Information about installing the **Log Parser** can be found at <u>http://www.microsoft.com/en-us/download/details.aspx?id=24659</u>

This query looks at the requested URLs ordered by time required to process the requests in descending order. This allows you focus on a rendering, or set of renderings in which to investigate.

Note that we are removing any URLs that include /sitecore/ to avoid looking at Sitecore client tools traffic.

To query the IIS Logs, looking for long running requests:

- 1. Launch the command line in a directory where the Log Parser is installed.
- 2. Run the following command --- change the #logs location# and #output file#:

```
logparser -i:IISW3C -o:CSV
"SELECT TO_TIMESTAMP(date, time) as [Timestamp], cs-uri-stem as [URI], c-ip AS [Client IP], time-
taken as [Time], sc-status as [Status]
INTO #output file#
FROM #logs location#
WHERE ((EXTRACT_EXTENSION(URI) = 'aspx' OR EXTRACT_EXTENSION(URI) = '') AND URI NOT LIKE
'%/sitecore%')
AND time-taken > 4000
AND Status = '200'
ORDER BY time-taken DESC"
```



2.2.5 Understanding the Results

Open the output file in Excel and look at the results in two parts. The first is a series of long running requests that are all coming from the same client IP address. For this purpose you can use sorting or filtering functionality in Excel. This could be an indication of in-house testing. Or could show the IP addresses with the slow or limited bandwidth. The second is a more typical set of results coming from external traffic to the website.

Results Part 1

Timestamp	URI	Client IP	Time	Status
27.01.2011 16:24:40	/Sverige.aspx	88.131.15.18	567,640	200
30.01.2011 08:35:30	/en/Norge.aspx	195.184.101.130	468,644	200
30.01.2011 08:35:30	/Danmark.aspx	195.184.101.130	468,363	200
30.01.2011 08:35:31	/Sverige.aspx	195.184.101.130	467,082	200
30.01.2011 08:35:31	/Sverige.aspx	195.184.101.130	466,504	200
30.01.2011 08:35:32	1	195.184.101.130	451,616	200
30.01.2011 08:35:33	/en/Norge.aspx	195.184.101.130	445,305	200
30.01.2011 08:35:32	/Danmark.aspx	195.184.101.130	444,945	200
30.01.2011 08:35:34	/Sverige.aspx	195.184.101.130	443,945	200
30.01.2011 08:35:33	/Sverige.aspx	195.184.101.130	443,492	200
30.01.2011 08:35:36	/UnitedKingdom.aspx	195.184.101.130	437,650	200
30.01.2011 08:35:36	1	195.184.101.130	437,462	200
30.01.2011 08:35:35	/Danmark.aspx	195.184.101.130	436,869	200
30.01.2011 08:35:35	/UnitedKingdom.aspx	195.184.101.130	436,791	200
30.01.2011 08:35:34	/en/Norge.aspx	195.184.101.130	436,588	200
30.01.2011 08:35:36	1	195.184.101.130	422,122	200

To see where the Client IP address is geo-located, go to http://ip2location.com/1.2.3.4, where 1.2.3.4 is replaced with the client IP address. For example, http://ip2location.com/195.184.101.130 would yield:

Image: State of the state o	IP Address	Country	Region	City	Latitude/ Longitude	ZIP Code	Time Zone	
Net Speed ISP Domain COMP RESULTMAKER A/S RESULTMAKER.COM 195.184.101.130 IDD Code Area Code Weather Station Map It 45 - DAXX0009 - COPENHAGEN Map It MCC MNC Mobile Brand -		E DENMARK	KOBENHAVN	COPENHAGEN	55.676294 12.568116	-	+01:00	
COMP RESULTMAKER A/S RESULTMAKER.COM 195.184.101.130 IDD Code Area Code Weather Station Map It 45 DAXX0009 - COPENHAGEN MCC MNC Mobile Brand		Net Speed		ISP		Domain		
195.184.101.130 IDD Code Area Code Weather Station Map It 45 DAXX0009 - COPENHAGEN MCC MNC Mobile Brand		COMP	RESULT	MAKER A/S	RES	ULTMAKER	.COM	
45 DAXX0009 - COPENHAGEN MCC MNC Mobile Brand	195.184.101.130	IDD Code	Area Code	•	Weather	Station		Map It
MCC MNC Mobile Brand		45	-	D	AXX0009 - C	OPENHAGE	N	
· · ·		MCC	MNC		Mobile B	Brand		
		-	-		-			

Note

This is also useful to track down possible network issues based on location.



Results Part 2

Timestamp	URI	Client IP	Time	Status
31.01.2011 10:13:48	/en/Company/Contact/Japan.aspx	173.203.158.156	5,170	200
31.01.2011 11:59:07	/News/RSS/Feeds/Denmark-News.aspx	193.3.234.5	5,155	200
30.01.2011 21:54:23	/Danmark.aspx	195.184.101.130	5,139	200
30.01.2011 11:54:36	/de/Hungary.aspx	78.46.71.246	5,121	200
27.01.2011 10:36:27	/en/Products/Resources/Tours.aspx	65.61.164.180	5,092	200
31.01.2011 10:17:45	/en/Customers/Selected-Customers.aspx	173.203.158.156	5,077	200
30.01.2011 11:57:48	/en/Solutions/Best-CMS-Solutions- Education.aspx	78.46.71.246	5,076	200
28.01.2011 01:09:32	/en/Japan.aspx	166.205.138.71	5,061	200
31.01.2011 10:14:02	/en/Products.aspx	173.203.158.156	5,046	200
29.01.2011 10:47:10	/en/Partners.aspx	65.61.143.45	5,030	200
27.01.2011 07:33:24	/products/resources/whitepapers/gartner-magic- quadrant.aspx	202.155.14.116	5,030	200
31.01.2011 10:14:10	/en/Partners/Hosting-Partners.aspx	173.203.158.156	4,999	200
31.01.2011 10:17:50	/en/Customers/Selected-Customers.aspx	173.203.158.156	4,983	200
28.01.2011 10:37:16	/en/Products/Industry-Commentary.aspx	64.39.4.224	4,983	200
29.01.2011 10:50:27	/en/Customers.aspx	65.61.143.45	4,983	200
31.01.2011 10:16:31	/Japan.aspx	173.203.158.156	4,983	200
30.01.2011 11:58:36	/en/News/NewsAndEvents.aspx	78.46.71.246	4,982	200
30.01.2011 11:56:21	/en/Company/Contact.aspx	78.46.71.246	4,966	200
28.01.2011 15:37:03	/Sverige/Partners/KnowlT.aspx	67.195.37.153	4,952	200
27.01.2011 10:35:45	/en/Partners/Become-Partner.aspx	65.61.164.180	4,952	200

After identifying URLs that either result from testing efforts or possible network issues due to geo-location, record the remaining.

If there is a number of URLs that are above the threshold, in our case 4 seconds, that occur during peak operation this could indicate a website that does not have enough capacity to handle the load.

2.2.6 Sitecore Recommendation

Sitecore recommends that the round trip time for any given .aspx page be under 4 seconds.



Report Findings

Record the Results

The peak hours of usage for the website are: _

Record any URLs that have a round trip time greater than 4 seconds. These URLs are used during the *Analyze Rendering Components* task.

Timestamp	URI	Client IP	Status Code	Time

There are requests that have exceeded the 4 second threshold ___ YES ___ NO

There is a large number of different URLs during peak operation hours ___ YES ___ NO.

There are requests that have exceeded the 4 second threshold = NO

OK. There are no request round trips that exceed 4 seconds.

There are requests that have exceeded the 4 second threshold = YES

Error. There are request round trips that have exceeded 4 seconds. Further investigation is required. Refer to the *Rendering Component Performance* task.

There is a large number of different URLs during peak usage hours = YES

Error. There is a large number of different URLs that have exceed the 4 second threshold during peak usage hours. This could be an indication of capacity problems and requires further investigation.



2.3 Rendering Performance

You can use the Sitecore stats page to collect information about rendering.

The stats page provides the following information about the various renderings used in a page, or throughout the website — depending when the stats page is observed.

- Rendering The name of the rendering.
- Site The name of the website that the information for the rendering is being collected.
- Count The number of times that the rendering has been called since the last time the stats page was reset.
- From cache The number of times the rendering was pulled from cache.
- Avg. time (ms) The average time taken to render to output.
- Avg. items The average number of items included in the rendering.
- Max. time The maximum amount of time taken to render the output.
- Max. items The maximum number of items included in the rendering.
- Total time The total amount of time taken for all instances of this rendering since the last stats page reset occurred.
- Total items The total number of items included in all instances of this rendering since the last stats page reset occurred.
- Last run This the last time that stats were collected.

2.3.1 Required Skills

• A working knowledge of the Sitecore stats.aspx page.

2.3.2 Symptoms

• Slow webpage round trip times.

2.3.3 Procedure to Use the Sitecore Stats Page

This procedure requires that you have permissions to access aspx pages in the /sitecore/admin folder. Also, we recommended that you open the stats page in a separate tab or browser window, so that you can keep it open while you use another tab or window to navigate through the website you are investigating.

To launch the stats page:

- 1. Launch either two web browser windows or tabs. We refer to these as the **stats** window and the **site** window.
- 2. In the stats window, go to http://<site>/sitecore/admin/stats.aspx.
- 3. The **stats** window shows information about the renderings that have been requested since the last time the stats page was reset.

If you want to start from scratch, click the refresh button — for example, if you want to view information regarding an individual web page you must clear out the stats page.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.



- 4. If you wish to view rendering stats that have been collected since the last reset, use the information presented.
- 5. If you wish to collect information about a single page, reset the stats page. In the **site** window, go to the website or webpage that you are going to collect stats about the renderings it includes.

Note

To collect information about caching and averages, make several calls to the page.

6. We recommended that you export the information in the stats page table to Excel, so that the information can be sorted for easier data analysis. To export the information, right click the table and then click **Export to Microsoft Excel**.

2.3.4 Understanding the Results

The following stats table has been exported into Excel. The information is sorted by Max. time, and some of the columns are hidden:

	А	В	С	D	E	G
1	Rendering	Site	Count	From cache	Avg. time (ms)	Max. time
2	Placeholder: content	nicam	20	0	185.3304	1670.1252
3	Sublayout: /layouts/Nicam/HomePageContent.ascx	nicam	2	0	1080.6819	1669.1879
4	Placeholder: rightcolumn	nicam	15	0	142.9569	1437.7889
5	/xsl/Nicam/NewsSpot.xslt	nicam	2	0	938.391	1419.1291
6	Sublayout: /layouts/Nicam/ThreeColumnContent.ascx	nicam	13	0	86.9166	362.6216
7	Placeholder: centercolumn	nicam	18	0	50.8885	328.2586
8	Sublayout: /layouts/Nicam/ProductForums.ascx	nicam	1	0	327.9864	327.9864
9	Sublayout: /layouts/Nicam/TwoColumnContent.ascx	nicam	5	0	82.8207	274.0745
10	Sublayout: /layouts/Nicam/ContactUsFormWrapper.ascx	nicam	1	0	260.099	260.099
11	FormRender1 (FormRender)	nicam	1	0	260.0562	260.0562
12	Placeholder: forum-content	nicam	1	0	178.4444	178.4444
13	Sublayout: /sitecore modules/Web/YAF/YAF_Forum.ascx	nicam	1	0	178.4045	178.4045
14	Placeholder: phxml	nicam	2	0	86.4851	162.7259
15	/xsl/FlashImageIterator/Flash_XMLOutput.xslt	nicam	2	0	86.4505	162.6944
16	/xsl/Nicam/Logo.xslt	nicam	21	7	12.5506	146.9511
17	Placeholder: frontpagebottomspotbar	nicam	2	0	84.1091	142.2216
18	Sublayout: /layouts/Nicam/Spots Three Column.ascx	nicam	4	0	42.0658	142.2035
19	Placeholder: spotbarcenter	nicam	4	0	33.869	121.6814
20	/xsl/Nicam/RotateSpots.xslt	nicam	6	0	32.8522	121.6473
21	/xsl/Nicam/FlexiblePersonalizationSpot.xslt	nicam	9	0	13.5927	111.2867
22	/xsl/Nicam/Top Menu.xslt	nicam	21	0	17.2713	106.0543
23	/xsl/Nicam/Product Catalog.xslt	nicam	12	0	17.6432	100.0303
24	Placeholder: frontflash	nicam	2	0	45.6854	89.0685

The following observations can be made from this table:

- All renderings that have a *Max time* in excess of 100ms need to be investigated to see if recommended coding practices have been followed.
- 0's in the *From cache* column indicate that Rendering (HTML Output) caching has not been configured.

2.3.5 Sitecore Recommendation

We recommend that renderings have a *Max time* of < 100ms, and that you enable and configure Rendering (HTML Output) caching.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.



Report Findings

Record the Results

Rendering *Max times* are < 100ms: __YES __NO There is several *From cache* values equal to 0: __YES __NO

Rendering *Max times* are < 100ms = YES:

OK. No rendering has a *Max time* > 100ms, per Sitecore recommended practices.

Rendering *Max times* are < 100ms = NO:

Error. There are renderings with a *Max time* > 100ms. Sitecore recommended practices are to keep rendering *Max times* less than 100ms.

There are several *From cache* values equal to 0 = NO:

OK. Rendering (HTML Output) caching has been enabled and configured per Sitecore recommended practices.

There are several *From cache* values equal to 0 = YES:

Error. Rendering (HTML Output) caching is either not enabled, and/or not configured. Sitecore recommended practices are to enable and configure Rendering (HTML Output) caching to improve website performance. For more information on how to enable and configure the Rendering caching, see the *Cache Configuration Reference* and the *Presentation Component Reference* manuals



2.4 Investigating Sitecore Memory Usage

The Sitecore system contains a series of performance counters that are logged to the Sitecore log files on a ten-minute interval:

- Process\Private Bytes
- Process\Virtual Bytes
- Process\Page File Bytes
- .net CLR Memory\# Bytes in all Heaps
- .net CLR Memory\% Time in GC
- .net CLR Memory\Large Object Heap size
- .net CLR Loading\Bytes in Loader Heap
- .net CLR Loading\Current Assemblies

If the Sitecore performance counters are not available, they can be downloaded from:

http://sdn.sitecore.net/upload/sdn5/faq/administration/sitecorecounters.zip.

This article explains possible problems with the counters and solutions for them: http://sdn.sitecore.net/Scrapbook/Working%20with%20Sitecore%20Performance%20Counters.aspx

The two counters that are of interest for this task are the *Process\Private Bytes* and the .net CLR Memory\# Bytes in all Heaps.

The *Process\Private Bytes* counter reports all memory that is exclusively allocated for the process — w3wp.exe — and can't be shared with other processes on the system. And the *.net CLR Memory\# Bytes in all Heaps* counter reports the combined total size of the Gen0, Gen1, Gen2, and large object heaps.

Typically the Private Bytes and # Bytes in all Heaps rise and fall at the same rate. If the Private Bytes is increasing, but the # Bytes in all Heaps remains stable, unmanaged memory is leaking. If both are increasing, and not being cleared, then there is a potential for a leak in the managed memory.

By using the Sitecore log files, *Sitecore Log Analyzer* and *Excel*, we graph these two counters to look for potential leaks.

2.4.1 Required Skills

- A working knowledge of the Sitecore Logs.
- A working knowledge of Sitecore Log Analyzer.
- A working knowledge of graphing with Microsoft Excel.

2.4.2 Symptoms

- OutOfMemory Exceptions
- IIS App Pool recycling
- Sluggish performance as memory usage increases.



2.4.3 Parsing the Sitecore Log(s) using Sitecore Log Analyzer

For more information about installing and setting up the **Sitecore Log Analyzer**, see <u>http://sdn.sitecore.net/Resources/Tools/Log Analyzer.aspx</u>

This task requires retrieving the values of the necessary health monitor counters from the log files and exporting them in CSV file.

The exported results are opened in **Excel** so that they can be graphed and compared.

Getting the Values of Process\Private Bytes and # Bytes in all Heaps Counters

The **Health Monitor** tab of **Sitecore Log Analyzer** allows you to see and export all health monitor counters of the parsed log files in a convenient format.

Date/Time	'Process\Private Bytes'	'.net CLR Memory\# Bytes in all Heaps'
10:05:58 18.12.2009	164,200,448	97,061,540
10:32:30 21.01.2010	141,377,536	52,807,420
11:23:51 21.01.2010	114,810,880	40,335,048
09:49:51 22.01.2010	131,223,552	39,760,008
12:18:35 22.01.2010	112,914,432	42,510,796
12:42:44 22.01.2010	116,015,104	41,054,552
12:52:49 22.01.2010	122,195,968	45,372,556
13:08:22 22.01.2010	138,801,152	66,553,120
14:30:35 22.01.2010	129,105,920	56,205,072
14:40:36 22.01.2010	144,379,904	62,259,972
14:59:44 22.01.2010	126,324,736	47.043.844
15:09:49 22.01.2010	128,786,432	51,487,452
15:58:23 22.01.2010	125,984,768	58,964,136
16:50:41 22.01.2010	170,356,736	47.046.864
17:00:46 22.01.2010	182,169,600	53,259,968
10:14:13 02.02.2010	122,175,488	43,359,768
10:30:22 02.02.2010	119,083,008	39,665,736

- 1. Launch Sitecore Log Analyzer.
- 2. Select logs for analyzing.
- 3. Click Analyze / Refresh.



4. Go to the **Health Monitor** tab and select the necessary counters — Process\Private Bytes and .NET CLR Memory\# Bytes in all Heaps.

Messages Audit Health Monitor Raw View Internal Program Log				
1027 CacheInstances 1027 CacheTotalCount 1027 CacheTotalSize	Date/Time	'Process\Private Bytes'	'.net CLR Memory\# Bytes in all Heaps'	
776 MemoryCommitted	10:05:58 18.12.2009	164,200,448	97,061,540	
776 MemoryUsed	10:32:30 21.01.2010	141,377,536	52,807,420	
201 PrivateBytes 138 'Process Private Bytes'	11:23:51 21.01.2010	114,810,880	40,335,048	
138 'Process\Virtual Bytes'	09:49:51 22.01.2010	131,223,552	39,760.008	
138 'Process\Page File Bytes'	12:18:35 22 01 2010	112,914,432	42,510,796	
91 '.net CLR Memory\# Bytes in all Heaps' (Showing on timeline)	12:42:44 22 01 2010	116.015.104	41 054 552	
91 '.net CLR Memoryl% Time in GC' 91 '.net CLR Memoryl% Time in GC'	12:52:49 22:01:2010	122 105 069	45 272 556	
91 '.net CLR Loading/Bytes in Loader Heap'	12:02:49 22:01:2010	122,193,300	66 552 120	
91 '.net CLR Loading\Current Assemblies'	13.06.22 22.01.2010	130,001,152	66,555,120	
	14:30:35 22:01:2010	129,105,920	56,205,072	
	14:40:36 22:01:2010	144,379,904	62,259,972	
	14:59:44 22.01.2010	126,324,736	47,043,844	
	15:09:49 22.01.2010	128,786,432	51,487,452	
	15:58:23 22.01.2010	125,984,768	58,964,136	
	16:50:41 22.01.2010	170,356,736	47,046,864	
	17:00:46 22.01.2010	182,169,600	53,259,968	
	10:14:13 02.02.2010	122,175,488	43,359,768	
Select All Deselect All	10:30:22 02.02.2010	119,083,008	39,665,736	
01.11 14.11 28.11 11.12 24.12 06.01 19.01 02.02	15.02 28.02 13.03	27.03 09.04 22	.04 05.05 18.05	01.06 14.06 27.06

- 5. Right click on the table and then click **Export Table To**, **CSV File**.
- 6. Give the csv file a meaningful name and open it in Excel.

Date/Time	'Process\Private Bytes'	'.net CLR Memory\# Bytes in all Heaps'	
10:05:58 18.12.2009	164,200,448	97,061,540	
10:32:30 21.01.2010	141,377,536	52 807 420	
11:23:51 21.01.2010	114 🔌 Export Tab	ole To 🔸 🗿 Cli	pboard
09:49:51 22.01.2010	131,223,552	39,760,008 🖬 CS	V File
12:18:35 22.01.2010	112,914,432	42,510,796	
12:42:44 22.01.2010	116,015,104	41,054,552	
12:52:49 22.01.2010	122,195,968	45,372,556	
13:08:22 22.01.2010	138,801,152	66,553,120	
14:30:35 22.01.2010	129,105,920	56,205,072	
14:40:36 22.01.2010	144,379,904	62,259,972	
14:59:44 22.01.2010	126,324,736	47,043,844	
15:09:49 22.01.2010	128,786,432	51,487,452	
15:58:23 22.01.2010	125,984,768	58,964,136	
16:50:41 22.01.2010	170,356,736	47.046.864	
17:00:46 22.01.2010	182,169,600	53,259,968	
10:14:13 02.02.2010	122,175,488	43,359,768	
10:30:22 02.02.2010	119,083,008	39,665,736	

Graphing the Results

There are several ways to create graphs in **Excel**. The following procedure describes how to create a line graph and compare the Process\Private Bytes results set to the .net CLR Memory\# Bytes in all Heaps.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.



1. In Excel, on the Insert tab, in the Charts group, click Line and then click 2D Line.

lome	Inse	rt	Page Layo	ut I	Formulas		Data	Review	N	Vie	W	Team	
		2	10		01	1		Ŵ				• 📥	:-:-
ble	Picture	Clip Art	Shapes	SmartAr	t Screen	shot	Column	Line		Pie	Bar	Area	Scatter
			Illustrati	ons				2-D	Line				
	-		f _x	Date	/Time				~1		1		
А			B	1					\sim		\simeq	\sim	
ne		138 '	Process\P	rivate	Bytes'	91 '	.net CLR						<u>, 1</u>
18.12	2.2009			164,2	200,448				ХI	12	≥ 1	1.2	40
21.0	1.2010			141,3	377,536			Ľ	_	Ľ	\simeq	~	20
21.0	1.2010			114,8	310,880			3-D	Line				48
22.0	1.2010			131,2	223,552								08
22.0	1.2010			112,9	914,432			K					96
22.0	1.2010			116,0	015,104								52
22.0	1.2010			122,1	195,968			đb	<u>A</u> II C	hart	Types		56
22.0	1.2010			138,8	301,152							66,553,	120
22.0	1.2010			129,1	105,920							56,205,	072
22.0	1.2010			144,3	379,904							62,259,	972

2. In the **Data** group, click **Select Data**.

If something was selected on the sheet, Excel attempts to create graph of it.

3. To clear the graph, remove any series and clear the Chart data range. The **Select Data Source** dialog box appears.

Chart data range:	
(Switch Row/Column
egend Entries (Series)	Horizontal (Category) Axis Labels
Add Edit X Re	move A V Edt
Hidden and Empty Cells	OK Cancel

- 4. Click Add.
- 5. For the first series, name the series **Private Bytes** and select the data range from the *Process\Private Bytes* column column B.

Series name:	
Private Bytes	📧 🛛 = Private Bytes
Series <u>v</u> alues:	
=Counters!\$B\$2:\$B\$111	= 164,200,448, 1
	OK Cancel



6. Set the Horizontal — x-axis — labels to the **Timestamp** column — column A — and click **Edit** and select column A from sheet 1:

Chart data range: =Counters!\$8	\$2:\$B\$111	
Ţ	Switch Row/Column)
egend Entries (Series)	Horizontal (Category) A	xis Labels
Add ZEdit XE	Remove 🔺 🔻 🗹 Edi <u>t</u>	
Private Bytes	1	*
	2	E
	3	
	4	
	5	-
Hidden and Empty Cells		OK Cancel
xis Labels	<u>8</u> X	
xis Labels Axis label range:		
xis Labels Axis label range: =Counters! \$A\$21;\$A\$111]	হ হ হ হ = 10:05:58 18.1	2

7. Click OK.

	(132)
	witch Row/Column
gend Entries (<u>S</u> eries)	Horizontal (Category) Axis Labels
Add	Edit
ivate Bytes	10:05:58 18.12.2009
	10:32:30 21.01.2010
	11:23:51 21.01.2010
	09:49:51 22.01.2010
	12:19:25 22:01 2010

8. Click **Add** and do the same for the second series — Bytes in all Heaps. Select the data range from the *.net CLR Memory\# Bytes in all Heaps* column — column C.



You do not need to set the Horizontal - x-axis - labels again.

Series name:	
Bytes in all Heaps	= Bytes in All H
Series <u>v</u> alues:	
=Counters!\$C\$2:\$C\$111	= 97,061,540, 52
	OK Cancel

Chart data range: =Counters!\$A\$2:\$C\$111		
Switch	h Row/Column	
gend Entries (Series)	Horizontal (Category) Axis Labels	
Add	Edi <u>t</u>	
rivate Bytes	10:05:58 18.12.2009	A
ytes in All Heaps	10:32:30 21.01.2010	
	11:23:51 21.01.2010	
	09:49:51 22.01.2010	
	12:18:35 22.01.2010	-

9. Click OK.



2.4.4 Understanding the Results (Graph)



Graph 1

- This graph illustrates an example of a potential memory leak in the managed memory space. The five big dips could represent the clearing of large Sitecore caches or app pool restarts due to either the app pool private memory limit being reached, or OOM exceptions. Between each dip you can see *Private Bytes* and *Bytes in all Heaps* are tending to increase in parallel. This could be a symptom of a memory leak, but it could also be normal reaction to increased load on the server requiring increased memory consumption and potential cache clearing. Further investigation would be required.
- Looking at our sample, we can see that both the *Private Bytes* and the *Bytes in all Heaps* are rising and falling at the same rate. It is OK for memory to raise to a stable point and remain there.

Note that our sample graph shows 5 events where memory is reset. Further investigation is explained later.

- If *Private Bytes* were rising, while *Bytes in all Heaps* are remaining constant, this would indicate that there is a potential leak in the unmanaged memory space and further investigation would be required.
- Peaks indicate memory allocation. When you move the mouse over a peak, information about the amount of memory allocated is displayed, as well as the time of the event:



Graph 2

• Detail view of *Private Bytes* spike, indicating peak memory usage prior to an app pool recycle or OOM exception.



• Valleys indicate when a garbage collection event has occurred. Moving the mouse over a valley provides information as to the memory still allocated, as well as the time that the event occurred:



Graph 3

- Details of a valley or GC event. Note, this is not an app pool recycle or OOM exception, but the normal activity of the garbage collector.
- By looking at the highest peaks and the lowest valleys, we can use this information to correlate information that we can obtain from the Windows event logs.
- The high points, just prior to the memory being reset can be compared to the private memory limit set for the App Pool in IIS Manager, select the appropriate App Pool, and select Advanced Settings.

Note

If there is no value set for the Private Memory limit on the App Pool, check the Sitecore logs for OutOfMemory Exceptions occurring during the same time frame.

In the following image, we can see that the App Pool is set to reset when memory reaches 1800000KB.

∃ Generate Recycle Event Log Entry	/	
Private Memory Limit (KB)	1800000	
Regular Time Interval (minutes)	1740	=
Request Limit	0	
Specific Times	TimeSpan[] Array	





Graph 4

- Details of a memory spike just prior to an app pool recycle or OOM exception. This information
 can be used to determine if the memory usage exceeds that of the available memory for the app
 pool.
- By comparing the low point that follows the peak with information from the Windows event logs, we can see that an App Pool recycle did indeed take place. This constant increase in both *Private Bytes* and *Bytes in all Heaps* followed by exceeding the memory limits set and a recycling of the App Pool could be a potential leak in the managed memory space. Further investigation would be



required.



Graph 5

• Details of a memory reset event. The date and time available can be used to correlate the information available from the Sitecore Logs and the Windows Event Logs to see if an OOM exception and / or an app pool recycling event has taken place.

Event 5117, WAS			×
General Details			
A worker process requested a recycl	with process id of '5004' servi e because it reached its priva	ng application po te bytes memory l	ol 'Sitecore.net_101201' has imit.
, Log Name:	System		
Source:	WAS	Logged:	12/23/2010 4:30:22 PM
Event ID:	5117	Task Category:	None
Level:	Information	Keywords:	Classic
User:	N/A	Computer:	WEB1D1.dk.sitecore.net
OpCode:	Info		
More Information:	Event Log Online Help		
•			

The time stamps match, indicating that the App Pool recycled.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.



2.4.5 Notes

For more information about .net memory usage and how to investigate it see:

- <u>http://msdn.microsoft.com/en-us/magazine/cc163491.aspx</u>
- http://msdn.microsoft.com/en-us/library/Ee817660(pandp.10).aspx



2.5 Sitecore Pipeline Profiling

By using the Sitecore pipeline profiling page, metrics related to the performance of core Sitecore pipelines can be monitored and collected. Only pipelines invoked by the

Sitecore.Pipelines.CorePipeline.Run() method are subject to profiling when pipeline profiling is enabled.

2.5.1 Required Skills

• A working knowledge of Sitecore pipelines and how they are used.

2.5.2 **Procedure to Use the Sitecore Pipeline Profiling Page**

This procedure requires that you have permissions to browse to aspx pages in the http://<site>/sitecore/admin directory.

To enable pipeline profiling, set the Pipelines.Profiling.Enabled setting to true in the web.config file or use the App Config\Sitecore.PipelineProfiling.config include file:

<setting name="Pipelines.Profiling.Enabled" value="true" />

Browse to the Sitecore pipeline profiling page:

http://<site>/sitecore/admin/pipelines.aspx

2.5.3 Overview

When pipeline profiling is disabled, the page displays a No data message, and instructions on how to enable profiling.

Pipeline Profiles A snapshot from the pipeline profiler.

Pipeline profiling is disabled. No data is currently available.

To enable pipeline profiling, in the 'web.config' file, set the value of the 'Pipelines.EnableProfiling' setting to 'True'.



When pipeline profiling is enabled, the page displays a list of the last used processors grouped by pipelines and usage statistics.

apshot from pipelines profiler.							
Refresh Reset							
Pipeline / Processor Name	#Calls	% Wall Time	Wall Time	Max Time	Avg Time	% CPU	Time / call
initialize	1		755.23	755.23	755.23		755.23
Sitecore.Pipelines.Loader.ShowVersion.Process	1	10.59	79.99	79.99	79.99	12.43	79.99
Sitecore.Pipelines.Loader.ShowHistory.Process	1	0.36	2.73	2.73	2.73	0.46	2.73
Citegere Binglings Londer SetClobals Process	1	0.61	4 62	4 62	4 62	0.78	4.62

2.5.4 Usage

The pipeline profiling page has two buttons: **Refresh** and **Reset**.

To retrieve the latest snapshot of profile counter values and render the page, click **Refresh**.

To resets the pipeline profiling counters, click **Reset**. After you click **Reset**, you must refresh the page to get an updated snapshot.

Pipeline Profiles A snapshot from the pipeline profiler.	- 1
Refresh	- 1
	_

2.5.5 Understanding the Results

Note

Certain pipelines are executed every time you click Refresh on the pipeline profiling page and therefore increment/impact the results for those pipelines for example, httpRequestBegin, preprocessRequest, renderLayout, getTranslation)

The pipeline profiling page contains the following information about Sitecore pipelines:

- The **# Executions** column provides the number of times the pipeline or processor has been executed over the lifetime of the profiler.
- The % Wall Time column is applicable to processors only and provides the ratio of execution time for a processor vs the execution time for the pipeline that invoked the processor. The values in this column are helpful in determining which processors consume the most time during pipeline execution.
- The **Wall Time** column provides the aggregate execution time for each pipeline or processor over the lifetime of the profiler. The values in this column are cumulative over time and are mostly useful for calculating average execution time.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.



- The Max Wall Time column provides the maximum time elapsed during the execution of a pipeline or processor. The values in this column represent the maximum execution time over the lifetime of the profiler and can be used to compare peak execution duration to average execution duration.
- The % CPU column is applicable to processors only and provides the ratio of CPU time used by a processor vs the CPU time used by the pipeline that invoked the processor.

Note

The % CPU metric is only measured and displayed when the Pipelines.Profiling.MeasureCpuTime setting is set to true in the web.config file (or via config include).
<setting name="Pipelines.Profiling.MeasureCpuTime" value="true" />

- The Time / Execution column provides the average execution time for each pipeline or processor over the lifetime of the profiler. The values in this column are valuable when monitoring pipeline/processor performance over a specific time period (e.g. during load or performance testing).

The top 3 worst performing processors — by execution/wall time — are marked with numbered icons.

preprocessRequest	1137		308.20	16.30	0.27		0.01<
${\it Sitecore. Pipelines. Preprocess Request. Suppress Form Validation. Process}$	1137	1.49	4.59	0.05	0.01<	2.24	0.01<
Sitecore.Pipelines.PreprocessRequest.NormalizeRawUrl.Process	1137	6.40	19.73	7.91	0.02	5.86	0.01<
Sitecore.Pipelines.PreprocessRequest.IIS404Handler.Process	1137	0.85	2.60	0.01	0.01<	1.20	0.01<
Sitecore.Pipelines.PreprocessRequest.WebDAVCustomHandler.Process	1137	86.40	266.29	7.96	0.23	85.35	0.01
Sitecore.Pipelines.PreprocessRequest.FilterUrlExtensions.Process	1137	3.93	12.12	0.31	0.01	4.25	0.01
Sitecore.Pipelines.PreprocessRequest.StripLanguage.Process	468	0.93	2.87	0.05	0.01<	1.10	0.01
1.00180080	*****			0.00			

Some processor names can contain more than 70 characters. In order to limit the processor name length in the results table, the full processor name is provided in a tooltip when you move the mouse over the processor name.

For compatibility and search/lookup needs, processor data tags have the following attributes:

- Title (compatibility) and pname (lookup) attribute contain full processor name.
- alt contains extended processor information for tooltip (html)

🛐 Sitecore.Automation.MarketingAutomation.Pipelines.LoadVisitor.LoadA	Sitecore.Automation.MarketingAutomation.Pipelines.LoadVisitor.LoadAutomations.Process
excludeRobots	% Wall Time: 1.20 Wall Time: 2.447758
2 Sitecore.Analytics.Pipelines.ExcludeRobots.TryObtainCachedResult.Pr	Max Time: 0.071993
Sitecore.Analytics.Pipelines.ExcludeRobots.CheckUserAgent.Process	% CPU cycles: 3.09
🛐 Sitecore.Analytics.Pipelines.ExcludeRobots.CheckIpAddress.Process	Time / call: 0.071993 CPU cycles: 6.32 G
Sitecore.Analytics.Pipelines.ExcludeRobots.AddResultToCache.Process	CPU cycles / call: 185.87 M



2.6 Sitecore Debugger

Sitecore provides a browser-based debugger which helps to locate problems associated with faulty or slow presentation components. The debugger helps developers analyze the output of both an entire page and the individual presentation components on a page.

2.6.1 Procedure to activate Sitecore Debugger

This procedure requires that:

• The Sitecore user has read access to the following item in the Core database:

/sitecore/content/Documents and settings/All users/Start menu/Right/Debug

- The allowDebug attribute for the current <site> is set to true in the web.config file or in the config include.

To activate the Sitecore Debugger:

- 1. Log in to the **Sitecore** Desktop.
- 2. Click **Sitecore**, and then click **Debug**

A new browser window or tab opens displaying the home page of the current website with the debugging ribbon across the top. You can use the button in the far right top of the ribbon to show or hide the debugging ribbon.

dit Debug -	Activate	Activate Save	Borders Information	Stop Off	
Mode	Profile	Trace	Rendering	Close	
`		•			
'					
-					
Siteco	ore				
welcome to S	bitecore				



2.6.2 Usage

In the Sitecore debugging ribbon, in the **Mode** group, click Edit to enable or disable inline editing. You can also switch between **Preview** and **Debug** mode.

	Save		Borders	%	
Edit Debug 👻	Activate Download	Activate Download	V Information	Log Off	
Mode	Profile	Trace	Rendering	Close	
•					
Siteco	ore				
Siteco	ore				
Siteco Welcome to	D re Sitecore				
Siteco Welcome to	D re Sitecore				
Siteco Welcome to	D re Sitecore				

In the **Profile** group, you can activate or de-activate Sitecore profiling for the current page. If Sitecore profiling is activated, you can save or download a profile report for the current page.

Edit Debug -	Activate Save	Activate Save	Borders Information Rendering	Log Off	
Mode	Pronie	Trace	Rendering	Close	E
2					
•					
Siteco	ore				
Welcome to	Sitecore				

A profile report is a high-level summary of the performance of a page. You can use the profile report to identify underperforming components within a page, and underperforming steps within those components.

When Sitecore profiling is activated, a profile report appears at the bottom of the page. The profile report attempts to identify hot spots — the worst-performing components, and components that accessed a large number of items. The profile report may contain error messages, such as, if layout details indicate to bind a presentation component to a placeholder that does not exist — which otherwise generates no



output to the page.

Sitecore Profile							
Hot Spots							
Most Time Taken:							
19.7% 3.148 ms Resolve device.							
19.2% 3.069 ms Render "Sublayout: /layouts/Sample Inner Sublayout.ascx".							
13.3% 2.125 ms Render "Placeholder: content".							
Most Items Read:							
13 Insert renderings into page.							
3 Expanding placeholder "/main"							
2 Processing rendering 'Sample Sublayout'							
Des Cla						_	
Profile							
Time Action	Total	Own 1	Items Read Data Cao	the Misses Data C	ache Hits Physi	cal Reads	
19.7% Resolve device.	3.148 ms	3.148 ms	0	0	0	0	
0.2% Resolve alias.	0.037 ms	0.037 ms	1	0	1	0	
0.4% Resolve current item.	0.058 ms	0.058 ms	1	0	1	0	
1.4% Resolve layout for "Home".	0.220 ms	0.220 ms	1	0	1	0	
0.0% Process page handlers.	0.005 ms	0.005 ms	0	0	0	0	
0.3% Check security access to page.	0.045 ms	0.045 ms	12	0	0	0	
1.0% Exceeding elsewholder "(main"	1.239 ms	0.207 ms	13	0	2010	0	
1.9% Expanding praceholder / main	0.050 ms	0.297 ms	3	0	2	0	
0.1% Expanding cubbyout "/byouts/Sample Subbyout accy"	0.361 ms	0.000 ms	2	0	2	0	
0.1% Loading user control "/avouts/Sample Sublavout ascy"	0.051 ms	0.020 ms	2	0	2	0	
0.5% Expanding placeholder "/main/centercolumn"	0.001 ms	0.001 ms	2	0	2	0	
O 0% Processing rendering 'Sample Inper Sublayout'	0.195 ms	0.005 ms	1	0	1	0	
0.1% Expanding subjayout "/layouts/Sample Inner Subjayout.ascx"	0, 190 ms	0.019 ms	1	0	1	0	
O.2% Loading user control "/layouts/Sample Inner Sublayout.ascx"	0.029 ms	0.029 ms	0	0	0	0	
0.1% Expanding placeholder "/main/centercolumn/banner"	0.015 ms	0.015 ms	0	0	0	0	
0.7% Expanding placeholder "/main/centercolumn/content"	0.127 ms	0.119 ms	1	0	1	0	
0.1% Processing rendering 'Sample Rendering'	0.008 ms	0.008 ms	0	0	0	0	
0.2% Inserting unused controls into the form element.	0.036 ms	0.036 ms	0	0	0	0	
0.2% Update browser caching headers.	0.037 ms	0.037 ms	0	0	0	0	
0.0% Checking for unused renderings.	0.003 ms	0.003 ms	0	0	0	0	
0.3% Render " (VisitorIdentification)".	0.044 ms	0.044 ms	0	0	0	0	
10.1% Render "Placeholder: main".	10.466 ms	1.615 ms	2	0	2	0	
10.4% Render "Sublayout: /layouts/Sample Sublayout.ascx".	8.851 ms	1.666 ms	2	0	2	0	
9.7% Render "Placeholder: centercolumn".	7.185 ms	1.547 ms	2	0	2	0	
19.2% Render "Sublayout: /layouts/Sample Inner Sublayout.ascx".	5.638 ms	3.069 ms	2	0	2	0	
0.2% Render "Placeholder: banner".	0.028 ms	0.028 ms	0	0	0	0	
13.3% Render "Placeholder: content".	2.541 ms	2.125 ms	2	0	2	0	
2.6% Render "/xsl/sample rendering.xslt".	0.416 ms	0.416 ms	2	0	2	0	
Total (including debug collection)	15.996 ms						

In the **Rendering** group, you can enable and disable the **Borders** and **Information options**. Enabling the **Borders** option displays borders that outline each presentation component on the current page. Enabling the **Information** option displays information icons — green triangles — for each presentation component on the current page.



Holding the mouse over an information icon displays information about that indivual presentation component inline in the page, including its profile, cache settings and output.

	Activate Download	99 Activate Download	Borders Information Rendering	Se Log Off	
D					
content_0					
Details Pro	ifile Cache Settings Output				
DataSourc Parameter	e: [current item] s: [none]				
WebContro	I: Sitecore.Web.UI.WebControls.Xs	irile			
E					

Note

When you select the Information checkbox, Sitecore invokes each presentation component on each page request and never retrieves the output of a presentation component from cache. To debug presentation component caching, do not select the **Information** checkbox.

In the **Trace** group, you can activate or de-activate Sitecore tracing for the current page. If Sitecore tracing is enabled, you can save or download a trace report for the current page. A trace is a low-level description of the steps involved in the page generation process.

Edit Debug -	Activate	Activate	Borders	See	
	FIONE		Kendening	Close	
9					
•					
Siteco	re				
Welcome to Sit	tecore				

If Sitecore tracing is activated, the trace report appears at the bottom of the page. You can use the trace report to determine whether or not Sitecore retrieved the output of each presentation component from cache as well as the order in which components were added to the page and the time needed to render



each component.

Sitecore Trace		
Type Action	Elapsed since last entry	Elapsed since start
Starting trace.	0.00 ms	0.00 ms
Current site is "website".	0.01 ms	0.01 ms
Current domain is "extranet".	0.01 ms	0.03 ms
Current language is "en".	0.02 ms	0.05 ms
Profiling is active.	0.01 ms	0.05 ms
Device set to "Default".	2.31 ms	2.36 ms
Current item is "/sitecore/content/home/".	4.30 ms	6.67 ms
Current layout is "Sample Layout".	0.64 ms	7.31 ms
Using physical layout "/layouts/Sample Layout.aspx".	0.03 ms	7.34 ms
Checking security for current user "sitecore\admin".	2.29 ms	9.62 ms
Access granted as the current user "sitecore\admin" has read access to current item.	0.03 ms	9.65 ms
Inserting rendering "Sample Sublayout" into page.	1.25 ms	10.90 ms
Inserting rendering "Sample Inner Sublayout" into page.	0.02 ms	10.92 ms
Inserting rendering "Sample Rendering" into page.	0.01 ms	10.93 ms
Adding Http headers to disable caching.	2.71 ms	13.64 ms
Adding Http header to indicate last modification. Date: 6/18/2009 12:53:23 PM.	0.02 ms	13.66 ms
Starting rendering " (VisitorIdentification)".	1.80 ms	15.46 ms
Elapsed time since start render start: 0.042 ms.	9.65 ms	25.10 ms
Collected debug information	3.52 ms	28.62 ms
Finished rendering " (VisitorIdentification)" .	0.61 ms	29.23 ms
Starting rendering " (WebEditRibbon)".	0.11 ms	29.34 ms
Elapsed time since start render start: 150.586 ms.	150.60 ms	179.94 ms
Finished rendering " (WebEditRibbon)" .	0.04 ms	179.98 ms
Starting rendering "Placeholder: main".	0.01 ms	179.99 ms
Starting rendering "Sublayout: /layouts/Sample Sublayout.ascx".	1.83 ms	181.82 ms
Starting rendering "Placeholder: centercolumn".	1.55 ms	183.37 ms
Starting rendering "Sublayout: /layouts/Sample Inner Sublayout.ascx".	1.61 ms	184.98 ms
Starting rendering "Placeholder: banner".	1.48 ms	186.46 ms
Elapsed time since start render start: 0.015 ms.	1.49 ms	187.95 ms
Collected debug information	0.07 ms	188.02 ms



Chapter 3

Search Diagnostics

The tools and procedures described in this chapter are intended to help identify, troubleshoot, and resolve Sitecore search-related issues. Utilizing the outlined techniques can also provide you with actionable information related to improving search performance.

This chapter contains the following sections:

- FillDB
- Verbose Logging
- LingScratchPad



3.1 FillDB

FillDB is a tool that allows Sitecore developers to rapidly fill databases with sample data.

The procedures described in this section are complementary to CMS tuning and diagnostics. By tuning the Sitecore CMS before and after data generation and running the diagnostic procedures before and after data generation, performance metrics can be compared across different-sized data sets. Developers and system administrators can use these metrics to identify potential bottlenecks in solution or system architecture.

3.1.1 Generating Items Using the FillDB Page

This perform this procedure, you must have permission to browse the aspx pages in the http://<site>/sitecore/admin folder.

Important

Sitecore does not recommend using the FillDB page to create sample items in any production database. You should use a test database or temporary database for sample items.

The sample items generated by the FillDB page are based on the *Sample Item* template that is found in standard Sitecore installations and also make use of the **Title** and **Text** fields in that template. The default *Folder* template is also used to create the hierarchy in which the sample items are stored. It is important that these templates exist in your Sitecore instance prior to using the FillDB process.

To use FillDB:

- 1. Locate the ItemGenerator.sql file in the /sitecore/admin/SqlScripts directory of your website.
- 2. Execute the SQL script in this file against the Sitecore Master database into which you'd like to insert sample items.

Create a directory called *data* in your website root directory and then create a directory within the *data* directory named *words* — c:\sitename\Website\data\words.

3. In the data\words directory, place some .txt files that contain large bodies of text.

Note

Project Gutenberg provides a convenient resource for downloading free text-based versions of books - http://www.gutenberg.org/

- 4. Browse to the FillDB page http://<site>/sitecore/admin/FillDB.aspx.
- 5. In the **Parent Guid** field, enter the ID of the item in your content tree under which you'd like to create the sample items. The default value provided in the field belongs to the standard Sitecore home item /sitecore/content/Home.

Note

The sample items generated by the FillDB page are placed in an auto-generated hierarchy of folders and items with no more than a 100 items in any given branch.

- 6. In the **Database Name** field, enter the name of the Sitecore database into which you'd like to insert sample items. The name should match the id attribute of a database that is defined in the databases section of the web.config file. The default value for this field is master.
- 7. In the Number of items field, specify the number of sample items that you want to create.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2013 Sitecore. All rights reserved.



8. Click Go! To create the items.

After the sample items have been created, you see a message indicating the time it took to create the items.

9. When the items have been created, clear your website cache.

To clear the website cache:

• Browse to the Sitecore admin cache page:

http://<site>/sitecore/admin/cache.aspx

On this page, click **Clear all** to clear all the Sitecore caches.

or

o Reset IIS.

or

• Recycle application pool for this website.



3.2 Verbose Logging

Verbose search logging is designed to facilitate search index configuration and provide insight in search troubleshooting scenarios. For example, if a particular item is not getting indexed, the VerboseLogger can provide more context and help you resolve the problem.

Important

Only enable the VerboseLogger component in special circumstances and never run it for long periods in a production environment. Otherwise, this would result in an extremely large log file, which may have performance implications.

To enable verbose search logging

- 1. Edit the App Config/Include/Sitecore.ContentSearch.config file
- 2. Set the ContentSearch.EnableSearchDebug setting to true:

<setting name="ContentSearch.EnableSearchDebug" value="true" />

3. Rename the

App_Config/Include/Sitecore.ContentSearch.VerboseLogging.config.example file to Sitecore.ContentSearch.VerboseLogging.config

Tha is — remove the .example file extension.

Alternatively, in the web.config file, add or patch the ContentSearch.VerboseLogging setting

<setting name="ContentSearch.VerboseLogging" value="true" />

4. In the web.config file, for the Sitecore search log, set the log4net logging level value to DEBUG:

```
<log4net>
<log4net>
<level value="DEBUG" />
<appender-ref ref="SearchLogFileAppender" />
</logger>
</log4net>
```



3.3 LinqScratchPad

Sitecore 7.0 includes a LINQ search API which provides convenient index searching capabilities using standard LINQ queries and syntax. LinqScratchPad is a tool that gives Sitecore developers the ability to execute LINQ search queries quickly and easily for testing and evaluation purposes.

3.3.1 Required Skills

• A working knowledge of Sitecore search queries and LINQ syntax.

3.3.2 Usage

To use the LingScratchPad tool you must have permission to browse to the aspx pages in the http://<site>/sitecore/admin directory.

Browse to the LingScratchPad page:

http://<site>/sitecore/admin/LinqScratchPad.aspx

You are presented with an interface like this:

<pre>using System: using System.ling; using System.ling; using System.ling; using Sistecce.ContentSearch.SearchTypes; using Sistecce.Bockets.Extensions; using Sistecce.Bockets.SearchTypes; using Sistecce.Bockets.SearchTags; using Sistecce.Bockets.SearchTags; using Sistecce.Bockets.SearchTags; using Sistecce.Bockets.SearchTags; using Sistecce.Bockets.SearchTags; using Sistecce.Bockets.SearchTags; using Sistecce.Bockets.SearchTags; using Sistecce.Bockets.SearchTags; using Sistecce.BotketSister using Sistecce.BotketSister using Sistecce.BotketSister using Sistecce.Sister; using (var context = SearchManager; {</pre>		
<pre>using System Collections.Generic; using System.Lindp: using System.Lindp: using System.Neb; using Sitecore.ContentSearch.SearchTypes; using Sitecore.Bouckets.SearchTypes; using Sitecore.Bouckets.SearchTypes; using Sitecore.Bouckets.Search.Tag; using Sitecore.ContentSearch; using Sitecore.ContentSearch.Utilities; using Sitecore.Deta.Fields; using Sitecore.Bote.Sites; using Sitecore.Bote.Sites; using Sitecore.Sites; using Constants = Sitecore.ScontentSearch.SearchManager; namespace Test { class Program { public static Sitemarable<searchmanager.getindex("sitecore_master_index").createsearchcor< th=""><th>1</th><th>using Sustam:</th></searchmanager.getindex("sitecore_master_index").createsearchcor<></pre>	1	using Sustam:
<pre>using System. Displacences, g using System. Link; using System. Neb; using Sistecce. Bockets. Extensions; using Sistecce. Bockets. SearchTypes; using Sistecce. Bockets. SearchTypes; using Sistecce. Bockets. SearchTypes; using Sistecce. Bockets. SearchTypes; using Sistecce. ContentSearch. Tag; using Sistecce. ContentSearch. Utilities; using Sistecce. Data. Fields; using Sistecce. Data. Fields; using Sistecce. Data. Fields; using Sistecce. Bata. Stams; using Sistecce. Bata. Stams; using Sistecce. Search; using Sistecce. Search; using Sistecce. Search; using Sistecce. Data. Then; using Sistecce. Search; using Sistecce. ContentSearch.SearchManager; namespace Test { class Frogram { public static Stopwatch stopWatch = new Stopwatch(); public static string RunTimer(string str) { return context. GetQueryble<searchresultitem>().Take(10).ToList(); } public static string RunTimer(string str) { return stopWatch.ElepsedWilliseconds.ToString(); } }</searchresultitem></pre>	2	using System Collections Comprise
<pre>d using bisection: using bisecter.line; using Sirecter.ContentSearch.SearchTypes; using Sirecter.Buckets.Search.Teaffaces; using Sitecter.Buckets.Search.Teaffaces; using Sitecter.Buckets.Search.Teaffaces; using Sitecter.Buckets.Search.Teaffaces; using Sitecter.Buckets.Search.Utilities; using Sitecter.ContentSearch.Utilities; using Sitecter.Btat.Fields; using Constants = Sitecter.ContentSearch.SearchManager; namespace Test { class Program {</pre>	3	using System Discontinues.
<pre>build given Networks Strees John Search Types; using Sitecore Buckts. Entersions; using Sitecore Buckts. Search Tags; using Sitecore Buckts. Search Tags; using Sitecore. ContentSearch, Tags; using Sitecore. ContentSearch; using Sitecore. ContentSearch; using Sitecore. ContentSearch; using Sitecore. Data; using Sitecore. Data; using Sitecore. Data; using Sitecore. Bata; Teads; using Sitecore. Bata; using Sitecore. Site; using Sitecore; using Sitecore; using Sitecore; using Constants = Sitecore. Suckets. Util. Constants; using Sitecore; using Sitecore; using Sitecore; using Constants = Sitecore. ContentSearchManager; namespace Test { class Program {</pre>	4	using System Ling:
<pre>using System.etc; using Steeres.ContempSearch.SearchTypes; using Steeres.Buckets.Extension; using Steeres.Buckets.SearchTypes; using Steeres.Buckets.SearchTypes; using Steeres.Buckets.SearchTypes; using Steeres.ContempSearch.Tag; using Steeres.ContempSearch.Utilities; using Steeres.Bata.Itea; using Steeres.Bata.Itea; using Steeres.Bata.Itea; using Steeres.Bata.Itea; using Steeres.BearLivGode; using Constants = Sitecore.Buckets.Util.Constants; using Steeres.Teas: using Constants = Sitecore.Buckets.Util.Constants; using (var context = SearchManager;</pre>	5	using System. Jung,
<pre>vising SiteCore LonenteesEntermings; using SiteCore Buckets Interfaces; using SiteCore Buckets Search. using SiteCore Configuration; using SiteCore.ContentSearch.Usities; using SiteCore.Deta.Eleds; using SiteCore.Deta.Eleds; using SiteCore.Eleds.Temm; using SiteCore.Sites; using (var context SearchSearchSearchSearch(); public static StopWatch.SiteSearchResultItem>Main(string str) { return stopWatch.SitepedMilliseconds.ToString(); } } } Keest Clear Run</pre>	6	using System. web,
<pre>vising Diffecter Buckets. Literations; using Siteccre Buckets. Search.7 using Siteccre Buckets. Search.7 using Siteccre ContentBearch. Using Siteccre ContentBearch. Using Siteccre Data. Using Siteccre Data. Using Siteccre Data. Using Siteccre Data. Using Siteccre Backets. Using Siteccre. SecurityModel; using Siteccre. New; using Siteccre. Using Siteccre. Using Siteccre. Using Siteccre. SecurityModel; using Siteccre. ContentBearch. SecurityModel; using Siteccre. Using Siteccre. ContentBearch. SecurityModel; using Siteccre. ContentBearch. SecurityModel; using Siteccre. ContentBearch. SecurityModel; using Siteccre. ContentBearch. SecurityModel; using Siteccre. ContentBearch. SecurityModel; using Constants = Siteccre. Buckets. Util. Constants; using Security using Constants = Siteccre. ContentBearch. SecurityModel; using (var context = SecurchManager; { class Program { creturn context. SecUrityModel: creturn context. SecUrityModel: creturn scoWatch. StapsedMilliseconds. ToString(); } } Meset Clear Run</pre>		using Sitecore.Contentsearch.Search.Ypes;
<pre>using Difects_Buckets_Search: using Sifects_Buckets_Search.Tags; using Sifects_ContentSearch.Usilties; using Sifects_ContentSearch.Usilties; using Sifects_ContentSearch.Usilties; using Sifects_Data.Items; using Sifects_Data.Items; using Sifects_Sites; using Sifects_Sites; using Sifects_Sites; using Sifects_Sites; using Sifects_Sites; using Sifects_Sites; using Sifects_Sites; using SearchWath; using Sifects_Sites; using (var context_SearchManager.GetIndex("sifects_master_index").CreateSearchCor { using Sifects_Sites; } } } Sifects</pre>	/	using Sitecore.suckets.Extensions;
<pre>9 Using Siteore.Buckets.Search.; 10 Using Siteore.Buckets.Search.; 11 Using Siteore.Buckets.Search.Utilities; 12 Using Siteore.Data; 13 Using Siteore.Data; 14 Using Siteore.Data; 15 Using Siteore.Data; 16 Using Siteore.Bata.Fields; 17 Using Siteore.Bata.Fields; 18 Using Siteore.Bata.Fields; 19 Using Siteore.SeaurityModel; 19 Using Vate static Stopwatch stopWatch = new Stopwatch(); 10 public static Innuesable<gearchmeaultitem> Main(string str) 10 { 11 Using (var context = SearchManager.GetIndex("sitecore_master_index").CreateSearchCor 12 { 13 Using (var context.GetQueryable<gearchresultitem>().Take(10).ToList(); 13 } 14 { 14 14 15 Using Siteore Static string RunTimer(string str) { 15 Jubic static string RunTimer(string str) { 16 Jubic static string RunTimer(string str) { 17 Teturn stopWatch.ElepsedMilliseconds.ToString(); 18 } 19 } 10 10 Jubic Run 11 11 12 Jubic Run 12 Jubic Run</gearchresultitem></gearchmeaultitem></pre>	8	using Sitecore.Buckets.Interfaces;
<pre>10 Using Sitecore.Buckets.Search.Tags; 11 using Sitecore.Configuration; 12 using Sitecore.ContentSearch; 13 using Sitecore.ContentSearch.Utilities; 14 using Sitecore.Data.Fields; 15 using Sitecore.Bata.Teams; 16 using Sitecore.Bata.Teams; 17 using Sitecore.Bata.Teams; 18 using Sitecore.Search.Util.Constants; 19 using Sitecore.Sites; 19 using Sitecore.Sites; 10 using Sitecore.Sites; 10 using Sitecore.Suckets.Util.Constants; 10 using Sitecore.Sites; 11 using Sitecore.Sites; 12 using Sitecore.Sites; 13 using Sitecore.Sites; 14 using Sitecore.Sites; 15 using Sitecore.Sites; 16 using Sitecore.Sites; 17 using Sitecore.Sites; 18 using Sitecore.Sites; 19 using Sitecore.Sites; 19 using Sitecore.Sites; 10 using Variants = Sitecore.ContentSearch.SearchManager; 10 namespace Test { 11</pre>	9	using Sitecore.Buckets.Search;
<pre>using Sitecore.Configuration; using Sitecore.ContentSearch.Ublities; using Sitecore.Data: using Sitecore.Data.Fields; using Sitecore.Data.Items; using Sitecore.SecurityModel; using Vare Sitecore.SecurityModel; using Vare static Stopwatch stopMatch = new Stopwatch(); public static IEnumerable<searchmanager.getindex("sitecore_master_index").createsearchcor { using (var context = SearchManager.GetIndex("sitecore_master_index").CreateSearchCor { return context.GetQueryable<searchresultitem>().Take(10).ToList(); } } } ublic static string RunTimer(string str) { return stopMatch.ElapsedMilliseconds.ToString(); } } } Reset Citear Run</searchresultitem></searchmanager.getindex("sitecore_master_index").createsearchcor </pre>	10	using Sitecore.Buckets.Search.Tags;
<pre>using Siteore.ContentSearch; using Siteore.ContentSearch; using Siteore.Data; using Siteore.Data.Tems; using Siteore.Bata.Tems; using Siteore.Store.Store; using Siteore.Store; using Siteore.Store; using Siteore.Store; using Siteore; using Siteore; using Siteore; using Siteore; using Siteore; using Siteore; using Constants = Siteore.ContentSearch.SearchManager; namespace Test { class Program { public static Stopwatch stopWatch = new Stopwatch(); public static ISnumerable<searchresultitem> Main(string str) { using (var context = SearchManager.GetIndex("siteore_master_index").CreateSearchCor</searchresultitem></pre>	11	using Sitecore.Configuration;
<pre>using Sitecore.ContentSearch.Utilities; using Sitecore.Data.Fields; using Sitecore.Data.Fields; using Sitecore.Sites; using Sitecore.Sites; using Sitecore.Neb; using Sitecore.Neb; using Sitecore.Sites; using Constants = Sitecore.ContentSearch.SearchManager; class Program {</pre>	10	using Sitecore.ContentSearch;
<pre>13 using Sitecore.Data.Fields; 14 using Sitecore.Data.Items; 15 using Sitecore.Bota.Items; 16 using Sitecore.SecurityModel; 17 using Sitecore.SecurityModel; 18 using Sitecore.Web; 19 using Sitecore.Web; 20 using Constants = Sitecore.Buckets.Util.Constants; 21 using Constants = Sitecore.ContentSearch.SearchManager; 22 using SearchManager = Sitecore.ContentSearch.SearchManager; 23 using SearchManager = Sitecore.ContentSearch.SearchManager; 24 namespace Test { 25 class Program { 26</pre>	12	using Sitecore.ContentSearch.Utilities;
<pre>14 using Sitecore.Brat.Fields; 15 using Sitecore.Brat.Fields; 16 using Sitecore.Booklets. 17 using Sitecore.SecurityModel; 18 using Sitecore.SecurityModel; 19 using Sitecore.SecurityModel; 20 using Constants = Sitecore.Buckets.Util.Constants; 21 using Constants = Sitecore.ContentSearch.SearchManager; 22 using SearchManager = Sitecore.ContentSearch.SearchManager; 23 using SearchManager = Sitecore.ContentSearch.SearchManager; 24 namespace Test { 26 class Program { 27</pre>	13	using Sitecore.Data;
<pre>15 using Sitecore.Bobalization; 16 using Sitecore.Bobalization; 17 using Sitecore.BecurityModel; 18 using Sitecore.Web; 19 using Sitecore; 20 21 22 using Constants = Sitecore.Buckets.Util.Constants; 23 using SearchManager = Sitecore.ContentSearch.SearchManager; 24 namespace Test { 25 class Program { 26 class Program { 27 public static Stopwatch stopWatch = new Stopwatch(); 28 using (var context = SearchManager.GetIndex("sitecore_master_index").CreateSearchCor 29 { 30 creturn context.GetQueryable<searchresultitem>().Take(10).ToList(); 33 } 34 } 35 public static string RunTimer(string str) { 36 return stopWatch.BlapsedMilliseconds.ToString(); 39 } 40 41 * * * * * * * * * *</searchresultitem></pre>	14	using Sitecore.Data.Fields;
<pre>16 using Sitecore.Globalization; 17 using Sitecore.Sites; 18 using Sitecore.Sites; 19 using Sitecore; 20 using Constants = Sitecore.Buckets.Util.Constants; 22 using SearchManager = Sitecore.ContentSearch.SearchManager; 23 namespace Test { 24 namespace Test { 25 class Program { 26</pre>	15	using Sitecore.Data.Items;
<pre>using Sitecore.SecurityWodel; using Sitecore.Neb; using Sitecore.Web; using Sitecore; using Constants = Sitecore.ContentSearch.SearchManager; using SearchManager = Sitecore.ContentSearch.SearchManager; class Program { private static Stopwatch stopWatch = new Stopwatch(); public static IEnumerable<searchresultitem> Main(string str) { using (var context = SearchManager.GetIndex("sitecore_master_index").CreateSearchCor { return context.GetQueryable<searchresultitem>().Take(10).ToList(); } public static string RunTimer(string str) { return stopWatch.ElapsedMilliseconds.ToString();</searchresultitem></searchresultitem></pre>	16	using Sitecore.Globalization;
<pre>using Sitecore.Sites; using Sitecore; using Constants = Sitecore.Buckets.Util.Constants; using SearchManager = Sitecore.ContentSearch.SearchManager; anmespace Test { class Program { private static Stopwatch stopWatch = new Stopwatch(); public static IEnumerable<searchresultitem> Main(string str) { using (var context = SearchManager.GetIndex("sitecore_master_index").CreateSearchCor { return context.GetQueryable<searchresultitem>().Take(10).ToList(); } } ublic static string RunTimer(string str) { return stopWatch.ElapsedMilliseconds.IoString(); } } Reset Reset Clear Run</searchresultitem></searchresultitem></pre>	17	using Sitecore.SecurityModel;
<pre>18 using Sitecore.Web; 19 using Sitecore; 20 21 22 using Constants = Sitecore.Buckets.Util.Constants; 23 using SearchManager = Sitecore.ContentSearch.SearchManager; 24 namespace Test { 26 class Program { 27 public static Stopwatch stopWatch = new Stopwatch(); 28 { public static IEnumerable<searchresultitem> Main(string str) 29 { { using (var context = SearchManager.GetIndex("sitecore_master_index").CreateSearchCor 30 { return context.GetQueryable<searchresultitem>().Take(10).ToList(); 33 } } 34 public static string RunTimer(string str) { 36 return stopWatch.ElapsedMilliseconds.ToString(); 37 } 38 } 40 41 </searchresultitem></searchresultitem></pre>	17	using Sitecore.Sites;
<pre>19 19 19 10 19 10 10 10 10 10 10 10 10 10 10 10 10 10</pre>	18	using Sitecore.Web;
<pre>20 21 22 23 24 using Constants = Sitecore.Buckets.Util.Constants; 23 24 namespace Test { 25 26 26 27 27 28 29 29 29 20 20 20 20 20 20 20 20 20 20 20 20 20</pre>	19	using Sitecore;
<pre>21 22 using Constants = Sitecore.Buckets.Util.Constants; using SearchManager = Sitecore.ContentSearch.SearchManager; 23 24 25 26 class Program { 27 28 29 29 29 29 29 20 20 20 20 20 20 20 20 20 20 20 20 20</pre>	20	
<pre>using Constants = Sitecore.Buckets.Util.Constants; using SearchManager = Sitecore.ContentSearch.SearchManager; namespace Test { class Program { private static Stopwatch stopWatch = new Stopwatch(); public static IEnumerable<searchresultitem> Main(string str) { using (var context = SearchManager.GetIndex("sitecore_master_index").CreateSearchCor { return context.GetQueryable<searchresultitem>().Take(10).ToList(); } public static string RunTimer(string str) { return stopWatch.ElapsedMilliseconds.ToString(); } } return tropWatch.ElapsedMilliseconds.ToString(); } Reset Clear Run </searchresultitem></searchresultitem></pre>	21	
<pre>using SearchManager = Sitecore.ContentSearch.SearchManager; namespace Test { class Program { private static Stopwatch stopWatch = new Stopwatch(); public static IEnumerable/SearchResultItem> Main(string str) { using (var context = SearchManager.GetIndex("sitecore_master_index").CreateSearchCor { using (var context.GetQueryable<searchresultitem>().Take(10).ToList(); } public static string RunTimer(string str) { return stopWatch.ElapsedMilliseconds.ToString();</searchresultitem></pre>	22	using Constants = Sitecore Buckets.Util.Constants:
<pre>24 namespace Test { class Program { private static Stopwatch stopNatch = new Stopwatch(); public static IEnumerable<gearchresultitem> Main(string str) { using (var context = SearchManager.GetIndex("sitecore_master_index").CreateSearchCor { return context.GetQueryable<searchresultitem>().Take(10).ToList(); } public static string RunTimer(string str) { return stopNatch.ElepsedMilliseconds.ToString(); } public static string RunTimer(string str) { return stopNatch.ElepsedMilliseconds.ToString();</searchresultitem></gearchresultitem></pre>	22	using SearchManager = Sitecore ContentSearch SearchManager:
<pre>24 25 26 class Program { 27 28 29 29 29 30 29 30 29 30 30 31 32 33 34 35 35 20 20 20 20 20 20 20 20 20 20 20 20 20</pre>	23	
<pre>25</pre>	24	namesnace Test /
<pre>26 27 28 29 30 30 30 31 32 33 34 35 35 36 4 36 37 37 38 39 39 40 41 Reset 26 class Program { 34 35 36 37 37 38 39 39 40 41 Reset 32 33 34 35 35 36 36 37 37 38 39 39 30 30 30 30 30 30 30 30 30 30 30 30 30</pre>	25	
<pre>27 27 28 29 29 30 4 29 4 29 4 29 4 29 5 29 5 29 5 29 5 29</pre>	26	class Drogram /
<pre>28</pre>	27	cruss regrum (
<pre>29</pre>	2.9	multic static IEnumerabla <gearchdesultitem> Main(string str)</gearchdesultitem>
<pre>29</pre>	20	public source induced be contractive substance and (source source)
<pre>30</pre>	29	l using (war context = SearchManager CetIndey("sitegore master indey") CreateSearchCor
<pre>31</pre>	30	(/ / / / / / / / / / / / / / / / / / /
<pre>32 33 34 35 public static string RunTimer(string str) { 36 37 39 39 40 41 </pre> <pre>Reset</pre> Clear Run	31	
33 34 35 36 37 38 39 40 41 Reset Reset Clear Run	32	return context.GetQueryable <searchkesuititem>().lake(10).lobist(),</searchkesuititem>
34 35 yublic static string RunTimer(string str) { 36 return stopWatch.ElapsedMilliseconds.ToString(); 37 38 39 40 41 Keset Clear Run	33	
35 public static string RunTimer(string str) { 36 return stopWatch.ElapsedMilliseconds.ToString(); 37 } 38 } 39 } 40 III 41 III Reset Clear Run	24	3
35 public static string kunimer(string str) { 36 return stopWatch.ElapsedMilliseconds.ToString(); 37 } 38 } 39 } 40	31	
36 return stopwatch.ElapsedMilliseconds.ToString(); 37 } 38 } 39 } 40	35	public static string Runlimer(string str) {
37 } 38 } 39 } 40	36	return stopWatch.ElapsedMilliseconds.ToString();
38 39 40 41 Keset Clear Run	37	
39 40 41 (m) Reset Clear Run	38	3
40 41 (m) Reset Clear Run	30	}
41 (m))	33	
41 (Reset Clear Run	40	
Reset Clear Run	41	+ III +
Reset Clear Run		
		Reset Clear Run



This interface allows you to write C# code that is compiled and executed. However, to ensure that the code executes properly for this page, you must follow a few conventions:

- Use the Test.Program type namespace Test, class Program
- The Program class must also contain two methods: Main and RunTimer.
 - The Test.Program.Main method is used to execute the query/code you are testing. The Test.Program.Main method should accept one argument of type string and return a value of type IEnumerable<SearchResultItem>
 - The Test.Program.RunTimer method is used to report the amount of time spent in executing your code from the Test.Program.Main method. The Test.Program.RunTimer method should accept one argument of type string and return a value of type string.

You will also notice a private Stopwatch field named stopWatch. This field can be used to measure the execution time of your query/code in the Test.Program.Main method, then used to report execution time via the Test.Program.RunTimer method. This field is not required, however, for proper execution.

Important

The Test.Program.Main method returns a value of type IEnumerable<SearchResultItem> that is used to populate a grid of results on the LingScratchPad page. It is important to note that returning a large result set, for example, thousands of results, will cause the page to render results slowly and may not be helpful when evaluating query execution time. It is recommended that you limit the number of results returned to avoid unnecessary delay.

For more information about using the LINQ to Sitecore search API, see the manual *Developer's Guide to Item Buckets and Search* on the SDN.