

Troubleshooting performance issues Framework NET Genium



netgenium.com



Content

1	Tas	sk Manager – CPU control	
	1.1	The CPU is permanently busy at 100%	4
	1.2	CPU usage oscillates around limit 70% and above	5
	1.2.	1 Examples of suboptimal algorithms	5
	1.3	CPU usage oscillates below 50%	5
2	Tas	sk Manager – memory check	6
2	2.1	The memory is permanently used at 100%	7
:	2.2	Memory usage is between 60% a 99%	8
2	2.3	Memory usage is below 50%	8
3	Pe	nding Windows operating system updates	9
4	De	bug Diagnostic Tool	10
5	Му	computer – Disk Check	11
6	Pe	rformance Manager – control of disk operations	14
7	Ru	nningQueries.exe	16
8	SQ	L Server – Activity Monitor	17
9	SQ	L Server – Top Queries by Total CPU Time	18
10	NE	T Genium logging and log evaluation	22
11	Edi	iting SQL queries and program code	23
	11.1	Indexes	23
	11.2	Joins	26
	11.3	The most common errors in database queries	27
	11.4	The most common errors when joining	27
	11.5	SQL Server Database Engine Tuning Advisor – SQL query debugging	28



1 Task Manager – CPU control

Windows Server 2016



Windows Server 2008





On the "Details" tab in the "Task Manager", sort the running processes in descending order according to the "CPU" column and find out which process is causing an unusual CPU usage.

👰 Task Manager	🖗 Task Manager 📃 🗖 🗙												
File Options View													
Processes Performance	Users	Details Services											
Name	PID	Status	User name	CPU	Memory (private wor	NP pool	Image path name						
System Idle Process	0	Running	SYSTEM	92	4 K	0 K							
📧 w3wp.exe	3068	Running	NETWORK SERVICE	04	2 433 628 K	862 K	C:\Windows\System32\inetsrv\w3wp.exe						
firebird.exe	1832	Running	SYSTEM	03	4 310 812 K	3 341 K	C:\Program Files\Firebird\Firebird_3_0\firebird.exe						
👰 Taskmgr.exe	12340	Running	Administrator	01	11 704 K	27 K	C:\Windows\System32\Taskmgr.exe						
System interrupts	-	Running	SYSTEM	00	0 K	0 K	-						
📧 w3wp.exe	3724	Running	NETWORK SERVICE	00	644 924 K	268 K	C:\Windows\System32\inetsrv\w3wp.exe						
📧 dwm.exe	4496	Running	DWM-2	00	14 232 K	42 K	C:\Windows\System32\dwm.exe						
📧 lsass.exe	624	Running	SYSTEM	00	35 404 K	26 K	C:\Windows\System32\Isass.exe						
📧 System	4	Running	SYSTEM	00	28 K	0 K	C:\WINDOWS\system32\ntoskrnl.exe						
📧 w3wp.exe	3096	Running	NETWORK SERVICE	00	603 336 K	99 K	C:\Windows\System32\inetsrv\w3wp.exe						
MsMpEng.exe	2336	Running	SYSTEM	00	165 108 K	75 K	C:\ProgramData\Microsoft\Windows Defender\platform\4.18.2007.8-0\M						
csrss.exe	152	Running	SYSTEM	00	1 380 K	20 K	C:\Windows\System32\csrss.exe						
📧 rdpclip.exe	8428	Running	Administrator	00	2 700 K	15 K	C:\Windows\System32\rdpclip.exe						
📧 svchost.exe	904	Running	NETWORK SERVICE	00	58 348 K	28 K	C:\Windows\System32\svchost.exe						
🐂 explorer.exe	9472	Running	Administrator	00	32 800 K	92 K	C:\Windows\explorer.exe						
HTOTALCMD64.EXE	8092	Running	Administrator	00	8 436 K	31 K	C:\Program Files\totalcmd\TOTALCMD64.EXE						
vm vmtoolsd.exe	7400	Running	Administrator	00	2 464 K	17 K	C:\Program Files\VMware\VMware Tools\vmtoolsd.exe						
🛇 ChangeService.exe	1784	Running	SYSTEM	00	22 240 K	37 K	D:\Services\ChangeService\ChangeService.exe						
孋 Ability Mail Server 2	2952	Running	SYSTEM	00	16 140 K	73 K	C:\Program Files (x86)\Code-Crafters\Ability Mail Server 2\Ability Mail Server 2						
Scr CRMService.exe	2124	Running	SYSTEM	00	17 884 K	38 K	D:\Services\CRMService\CRMService.exe						
📧 svchost.exe	1060	Running	SYSTEM	00	35 136 K	68 K	C:\Windows\System32\svchost.exe						
🔏 SearchIndexer.exe	6112	Running	SYSTEM	00	20 280 K	71 K	C:\Windows\System32\SearchIndexer.exe						
Service.exe	2180	Running	SYSTEM	00	15 932 K	36 K	D:\Services\PaymentService\PaymentService.exe						
SadMailService.exe	2052	Running	SYSTEM	00	15 740 K	29 K	D:\Services\BadMailService_OCD\BadMailService.exe						
SESLService.exe	2116	Running	SYSTEM	00	14 832 K	36 K	D:\Services\ESLService\ESLService.exe						
<			ш				>						
Fewer details							End task						

1.1 The CPU is permanently busy at 100%

- A Permanent CPU usage can indicate a web application, console application, or service loop that is caused by an error in the application source code.
- Permanent CPU load by the "w3wp.exe" process can indicate a cycle of the application pool or an error in one of the web applications. According to the PID of the "w3wp.exe" process, it is necessary to find the relevant application pool in IIS and identify the web applications that are served by this application pool.
- Permanent CPU usage by the "firebird.exe" process can indicate either poorly designed SQL queries that disproportionately use the Firebird database server, or a long-term increasing database server load that must be addressed by a hardware increase in the number of processor cores.
- Permanent CPU usage by the "sqlserver.exe" process can indicate either poorly designed SQL queries that disproportionately load the MSSQL database server, or a long-term increasing database server load that must be addressed by a hardware increase in the number of CPU cores.



1.2 CPU usage oscillates around limit 70% and above

- Higher CPU utilization by the "w3wp.exe" process may indicate suboptimal algorithms in some of the web applications. According to the PID of the "w3wp.exe" process, it is necessary to find the relevant application pool in IIS and identify the web applications that are served by this application pool.
- Higher CPU usage by the "firebird.exe" process can indicate either poorly designed SQL queries that disproportionately load the Firebird database server, or a long-term increasing database server load that must be addressed by a hardware increase in the number of CPU cores.
- Higher CPU usage by the "sqlserver.exe" process can indicate either poorly designed SQL queries that disproportionately use the MSSQL database server, or a long-term increasing database server load that must be addressed by a hardware increase in the number of CPU cores.

1.2.1 Examples of suboptimal algorithms

- For cycles that call an SQL query in each iteration the solution is to load all data in bulk before the for cycle itself, and then index this data into a "Dictionary" object. This procedure is described in detail in the separate "External functions" manual.
- Generating large files in PDF or XLSX format using "Aspose" libraries the solution is a hardware increase in the number of cores.
- High-resolution image processing the solution is a hardware increase in the number of cores.
- Non-optimal algorithms are often located in one of the external functions, or they can be caused by printing to printing templates. In both cases, it is important to check the script report (total_worker_time) and the external functions (total_worker_time), in which the long durations of the scripts or external functions are displayed at the top of the report.

1.3 CPU usage oscillates below 50%

This condition is normal.



2 Task Manager – memory check

Windows Server 2016



Windows Server 2008





On the "Details" tab in the "Task Manager", sort the running processes in descending order according to the "Memory (private working set)" column and find out which process is causing unusual memory usage.

🙀 Task Manager							X
File Options View							
Processes Performance	Users	Details Services					
Name	PID	Status	User name	CPU	Memory (private wor	NP pool	Image path name
firebird.exe	1832	Running	SYSTEM	05	4 319 344 K	3 349 K	C:\Program Files\Firebird\Firebird_3_0\firebird.exe
📧 w3wp.exe	3068	Running	NETWORK SERVICE	02	2 438 992 K	879 K	C:\Windows\System32\inetsrv\w3wp.exe
📧 w3wp.exe	3724	Running	NETWORK SERVICE	00	646 948 K	267 K	C:\Windows\System32\inetsrv\w3wp.exe
📧 w3wp.exe	3096	Running	NETWORK SERVICE	00	531 360 K	100 K	C:\Windows\System32\inetsrv\w3wp.exe =
MsMpEng.exe	2336	Running	SYSTEM	00	165 924 K	75 K	C:\ProgramData\Microsoft\Windows Defender\platform\4.18.2007.8-0\\
📧 svchost.exe	904	Running	NETWORK SERVICE	00	58 404 K	28 K	C:\Windows\System32\svchost.exe
Isass.exe	624	Running	SYSTEM	00	35 628 K	26 K	C:\Windows\System32\Isass.exe
📧 svchost.exe	1060	Running	SYSTEM	00	35 332 K	68 K	C:\Windows\System32\svchost.exe
📏 VisualSVNServer.exe	3632	Running	NETWORK SERVICE	00	33 116 K	52 K	C:\Program Files (x86)\VisualSVN Server\bin\VisualSVNServer.exe
🐂 explorer.exe	9472	Running	Administrator	00	32 864 K	92 K	C:\Windows\explorer.exe
📧 w3wp.exe	10464	Running	NETWORK SERVICE	00	28 328 K	63 K	C:\Windows\System32\inetsrv\w3wp.exe
🛇 CRMService.exe	2124	Running	SYSTEM	00	21 312 K	37 K	D:\Services\CRMService\CRMService.exe
饺 ChangeService.exe	1784	Running	SYSTEM	00	21 104 K	36 K	D:\Services\ChangeService\ChangeService.exe
🙈 SearchIndexer.exe	6112	Running	SYSTEM	00	18 984 K	71 K	C:\Windows\System32\SearchIndexer.exe
Service.exe	2180	Running	SYSTEM	00	16 768 K	36 K	D:\Services\PaymentService\PaymentService.exe
📧 dwm.exe	4496	Running	DWM-2	00	16 596 K	41 K	C:\Windows\System32\dwm.exe
Mail Server 2	2952	Running	SYSTEM	00	16 176 K	73 K	C:\Program Files (x86)\Code-Crafters\Ability Mail Server 2\Ability Mail S
😳 BadMailService.exe	2052	Running	SYSTEM	00	15 740 K	29 K	D:\Services\BadMailService_OCD\BadMailService.exe
SLService.exe	2116	Running	SYSTEM	00	14 832 K	36 K	D:\Services\ESLService\ESLService.exe
📧 inetinfo.exe	1048	Running	SYSTEM	00	12 536 K	16 K	C:\Windows\System32\inetsrv\inetinfo.exe
📧 svchost.exe	1348	Running	NETWORK SERVICE	00	12 444 K	37 K	C:\Windows\System32\svchost.exe
📧 dwm.exe	872	Running	DWM-1	00	12 292 K	18 K	C:\Windows\System32\dwm.exe
👰 Taskmgr.exe	12340	Running	Administrator	04	11 812 K	27 K	C:\Windows\System32\Taskmgr.exe
📧 svchost.exe	980	Running	LOCAL SERVICE	00	11 540 K	18 K	C:\Windows\System32\svchost.exe
Service.exe	2076	Running	SYSTEM	00	10 528 K	27 K	D:\Services\EETService\EETService.exe
<			III				>
Fewer <u>d</u> etails							<u>E</u> nd task

2.1 The memory is permanently used at 100%

- Permanent memory usage by the "w3wp.exe" process may indicate suboptimal algorithms or loading too much data from the database in one of the web applications. According to the PID of the "w3wp.exe" process, it is necessary to find the relevant application pool in IIS and identify the web applications that are served by this application pool. For all these applications, it is necessary to turn on database logging and memory consumption measurement in the NET Genia settings, see chapter "NET Genium logging and log evaluation".
- Permanent memory usage by the "firebird.exe" process can indicate either poorly designed SQL queries that disproportionately load the Firebird database server, or a long-term increasing database server load that must be addressed by a hardware increase in memory.
- Permanent memory usage by the "sqlserver.exe" process can indicate either poorly designed SQL queries that disproportionately load the MSSQL database server, or a long-term increasing database server load that must be addressed by hardware-increasing memory.
- The permanent memory usage of the "sqlserver.exe" process means, among other things, that one of the instances (usually the default one) does not have a set memory usage limit, which in turn prevents the operating system and other applications from running properly. Each instance of MSSQL should always have a memory usage limit set to a safe limit, such as only 80% from the total memory so that the operating system and other applications for their operation.



2.2 Memory usage is between 60% a 99%

- High memory usage by the "w3wp.exe" process may indicate that some of the web applications are loading too much data from the database. According to the PID of the "w3wp.exe" process, it is necessary to find the relevant application pool in IIS and identify the web applications that are served by this application pool. For all these applications, it is necessary to turn on database logging and memory consumption measurement in the NET Genia settings, see chapter "NET Genium logging and log evaluation".
- **1** It is a good idea to consider a hardware increase in memory primarily to run more memory-intensive tasks once.
- **1** You must check the MSSQL server instance settings to see if it has a memory usage limit.

2.3 Memory usage is below 50%

This condition is normal.



3 Pending Windows operating system updates

On the application and database server, check the queue for pending updates that need to complete the installation. These updates can wait in the background and take up memory for the entire server, even though in Task Manager it appears that the server has enough free memory. Consequently, it does not have the resources to start and run processes normally, and both common applications on the application server and the database server collapse. Typically, this condition on the database server is that the MSSQL instance allocates only 200MB, for example, even though it has a much higher limit or no limit on the allocated memory at all.



4 Debug Diagnostic Tool

- In critical situations, it is necessary to end individual processes in the "Task Manager". Before this step, it is useful to create a "memory dump" of the process, which can then be analyzed using the "Debug Diagnostic Tool" and can detect both looping algorithms and the reason for memory congestion.
- A simplified procedure for working with "memory dumps" is given in the file "Config\Tools\MemoryDumps.txt" every NET Genium:

1) Download and install Debug Diagnostic Tool v2 Update 2 $\,$

- (https://www.netgenium.com/download/DebugDiagx64.msi)
- 2) Locate memory dumps (C:\Users\abc\AppData\CrashDumps)
- 3) Run DebugDiag
- 4) Default Analysis / CrashHangAnalysis
- 5) Add Data Files
- 6) Start Analysis



5 My computer – Disk Check

▲ Incoming disk space often means irreversible data loss due to database consistency. The fastest way to free up disk space is to delete log files or database and file backups.



The "TDP x-Ray" program is an ideal tool for detailed analysis of disk occupancy by individual directories and files.



- The log files include both the IIS log files located in the default directory "C:\inetpub.\logs\LogFiles", so the log files of specific NET Genium the "Logs" directory of each NET Genium.
- O Database backups are often located in the "Backup" directory of each NET Genium, or in the "E:\BackupServer".
- ▲ By default, IIS has web application traffic logging turned on, which must be turned off as soon as possible. At the same time, it is important to delete existing logs, which are stored in the "C:\inetpub\logs\LogFiles". Use the following procedure to determine the exact location of these logs from the specific IIS settings.



Go to the IIS settings and select the highest node of your server/computer (for example "OFFICE" / "TOMAS-PC") so that the settings are overwritten to other child nodes, and select "Logging" / "Logging" in the "IIS" section.







A Then click "Disable" in the "Actions" panel on the right side of the window. This disables vehicle logging for your server/computer node.

Ale and a second	Actions
Logging	By Apply
Use this feature to configure how IIS logs requests on the Web server.	🖳 Cancel
One log file per	↑ Disable
Cite V	View Log Files
Site	🕡 Help
Log File	
Format:	
W3C V Select Fields	
Directory:	
%SystemDrive%\inetpub\logs\LogFiles Brow	vse
Encoding:	
UTF-8 \checkmark	
Log Event Destination	
Select the destination where IIS will write log events.	
l og file oply	
	¥
ξ.	>
🖺 Features View 💦 Content View	
unfig	G





6 Performance Manager – control of disk operations

On the "Performance" tab in the "Task Manager", click on the "Open Resource Monitor" link.







On the "Disk" tab in the "Resource Monitor", sort the disk activities in descending order according to the "Write" column and find out which files – Firebird or MSSQL databases – show unusual disk writes.

🔊 Resource Monitor									-		×	
File Monitor Help												
Overview CPU Memory	Disk 🛛 👔	Network										
Processes with Disk Activity							 ^ 	>	N	liews	-	^
Image ISDSService.exe System MrMpEng.gro	PID 10328 4							Disk		1 MB/se		
msmpEng.exe firebird.exe w3wp.exe perfmon.exe w3wp.exe	1832 3068 12884 3096							AMA.A.	MA			
<		Ш					>	b0 Seconds Disk 0 (C:) C	ueue Len	igth 0.0		
Disk Activity		110 KB/sec Disk I/C)	196 High	est Active Time					- 1		
lmage System System	PID 4 4	File D:\Firebird\HE C:\ProgramDat	Read (B/sec) 0 0	Write (B/sec) 18 993 14 170	Total (B/sec) 18 993 14 170	I/O Priority Normal Normal	Respo ^	A, AA		<u>,</u>		-
System System	4 4	D:\Firebird\SP C:\\$LogFile (NT	0	12 281 10 513	12 281 10 513	Normal Normal		Disk 1 (D:) C	ueue Len	igth 0.0	05	
System System firebird.exe	4 1832	D:\Services\ISD C:\ProgramDat D:\StarsSita (N	0	4 096 3 938	4 096 3 938	Normal Normal				A		
System	4	D:\Firebird\SL	0	2 730	2 730	Normal	>	Mr.A.		₿.A		
Storage							•	Disk 2 (E:) Q	ueue Len	gth		



7 RunningQueries.exe

The "RunningQueries.exe" application is located in the "bin" directory of each NET Genium, and is used to evaluate currently processed database queries. Running this application creates a "RunningQueries.htm" log file in the "Logs" directory, and opens this log file at the same time. The queries mentioned in this log file may reduce the performance of the database server. At the same time, they can be used to identify the database in which performance issues occur.

€ E:\NETG	enium\Logs\RunningQuerie:	s.htm - Internet Expl	orer											
$\bigcirc \bigcirc \bigcirc$	🕞 🔄 🖉 E:\NETGenium\Logs\RunningQueries 🔎 💽 🏀 E:\NETGenium\Logs\Running ×													
Server	Database	Program	LoginName	StartTime	Status	Query	KillQuery							
(local)	continous-beam	CZNG/.Net SqlClient Data Provider	DLUBAL\Service_NetGenium	13.8.2020 13:32:04	suspended	(@ng_opendate datetime)UPDATE ng_membercampaign SET ng_opendate = @ng_opendate WHERE id = 10010056	kill 86							
(local)	cal) continous-beam CZNG/.Net SqlClient Data Provider		DLUBAL\Service_NetGenium	13.8.2020 13:32:04	running	(@1 smallint)SELECT * FROM [scolumns] WHERE [id]=@1	kill 115							
(local)	SCVirtualManagerDB	CZNG/.Net SqlClient Data Provider	DLUBAL\Service_NetGenium	13.8.2020 13:32:04	running	SELECT * FROM scolumns WHERE id = 703	kill 115							



8 SQL Server – Activity Monitor

Start "SQL Server Management Studio" and click on the "Activity Monitor" icon.



On the "Recent Expensive Queries" tab, identify the list of recently run demanding database queries. These queries can also be used to identify the database in which performance issues occur.

Overview									\odot
% Processor Time (2%) 100 80 -80 60 -40 -20 0	Waiting Tasks (0)		10 Databa - 8 - 6 - 4 - 2 - 0	se I/O (1,9 MB/se	c) 10 	Batch Request	s/sec (170)	10000 -8000 -6000 -4000 -2000 -0	
Processes									\odot
Resource Waits									\odot
Data File I/O									\odot
Recent Expensive Queries									۲
Query 🗸	Executions/r	CPU (ms/sec)	Physical Reads/sec	Logical Writes/sec	Logical Reads/sec	Average Duration (ms)	Plan Count	Database Name	-
WITH merged_query_stats AS(SELECT	2	69	50	115	11739	2451	1	tempdb	
SELECT COUNT(*) FROM sinvalidlogins WHE	1520	28	0	0	9221	1	529	netgenium	
SELECT ng_timesheet.ng_date c23260, ng_ti	2	6	0	0	1029	27	1	netgenium	
SELECT ng_timesheet.ng_date c23260, ng_ti	2	6	0	0	1008	25	1	netgenium	
SELECT ng_timesheet.ng_date c23260, ng_ti	2	5	0	0	1001	26	1	netgenium	
SELECT susers.loginname c7, susers.ng_color	68	2	0	0	107	1	1	netgenium	
SELECT j1.ng_segmenttype c9088, j1.ng_color	2	2	0	0	55	60	1	netgenium	
SELECT @i = COUNT(") FROM @table1 t1 IN	497	1	0	0	49	0	1	netgenium	
SELECT * FBOM [scolumns] WHEBE [id]=@1	1977	1	0	0	127	0	3	netaenium	



9 SQL Server – Top Queries by Total CPU Time

Start "SQL Server Management Studio", right-click on the top node in "Object Explorer", and select "Reports / Standard Reports / Performance – Top Queries by Total CPU Time".



Microsoft SQL 9	Server Management Studio							
File Edit View	Project Debug Tools Window C	ommunity Help						
New Query	🗅 🔁 🔁 🌇 🕒 🐸 🗔 🥌 🛎	4 -						
Object Explorer		→ ╄ ×	CZNG - Activity Monitor					
Connect 🕶 📑 📑	= T <u>S</u>		Overview					
Image Image Image	Connect Connect Disconnect Register New Query Activity Monitor Start Stop Payse Resume Restart Deploy Data-tier Application Policies Facets Start PowerShell Refresh Properties	Standard Rep Performance Performance Memory Cor Activity - All Activity - All	CZNG - Activity Monitor Overview % Processor Time (4%) % Processor Time (4%) % Processes Resource Waits Data File I/O Recent Expensive Queries Query W/ITH merged_query_stats AS (SELEC select /*s.id. s./ *na_chanaeseament* // aports ************************************	00 0 0 0 0	Waiting Tasks (0)	CPU rd anges History s History n piption cking Transactions sors ursors sions sors sions sistors	Physical	Logica
Output Beady		Server Dash	hboard		Top Transactions Top Transactions Top Transactions Performance - B Performance - To Performance - To Performance - To Performance - To Service Broker S Transaction Log	s by Age s by Blocked Trans s by Locks Count atch Execution Sta object Execution St op Queries by Ave op Queries by Ave op Queries by Tot op Queries by Tot tatistics Shipping Status	atistics atistics arage CPU Time arage 10 al CPU Time at 10	

In this report, identify the list of database queries that are most burdensome on the database server. These queries can also be used to identify the database in which performance issues occur.





- Once the database in which the performance issues occur is identified, it is more convenient to run a specific NET Genium and display the "dm_exec_query_stats (total_worker_time)" report. This report displays an identical list of database queries that are most burdensome on the database server, and also offers the ability to identify the control that runs the database query ("…" link).
- SQL Server also offers a "Top Queries by Average CPU Time" report, which has the equivalent in NET Genium under the "dm_exec_query_stats (average_worker_time)" report.
- **1** In general, it is recommended to browse the reports in the following order:
 - dm_exec_query_stats (total_worker_time) In this report, both database queries that may have a reasonable query processing time but run too often and database queries that run at reasonable intervals but have a long processing time come first.
 - **dm_exec_query_stats (execution_count)** In this report, database queries that run frequently come first, so it is important to consider "average_worker_time" for these queries.
 - dm_exec_query_stats (average_worker_time) In this report, database queries come first, which have a long processing time of individual queries.
- A typical incorrectly designed query error that runs very often and does not have an index set on the "ng_url" column.



KMicrosoft SQL Server Management Studio					
<u>File Edit View Project Debug Iools Window Community H</u> elp					
🗄 🔔 New Query 📄 📸 📸 🛐 📑 💕 🛃 🥌 🜉 💂					
Object Explorer 🚽 🗸 🗸	Performance	2020 9:55 - CZNG			
Connect 🕶 📑 📑 👅 🐼	1 🧔 🖨				
 CZIVG (SQL Server 10.50.4000 - CZIVG(netgenium) Databases Security Server Objects Replication Management SQL Server Agent 	Perforr Time on CZNG This report ide executions. 1	nance - Top Queries by Total CPU at 26.8.2020 9:55:53 entities the queries currently residing in the plan cache that have consumed the most to bits data is aggregated over the lifetime of the plan in the cache and is available only for Total CPU Time	tal CPU time over plans currently i	r the course of the cache.	er "2008 R2
	8	Avg. CPU Time			
	6 (19 19 19 19 19 19 19 19 19 19 19 19 19 1		78	9	
	Query No.	Query Text	Database Name	Object ID	Total CPU Time (ms.)
	1	SELECT ng_internalid, ng_url, ng_type, ng_language FROM ng_languagedependenturl WHERE ng_url = @url			9 512 830,26
	2	SELECT ng_internalid, ng_url, ng_type, ng_language FROM ng_languagedependenturi WHERE ng_url = @url			6 882 270,40
	3	SELECT ng_internalid, ng_url, ng_type, ng_language FROM ng_languagedependenturi WHERE ng_url = @url			6 768 798,41
	4	B SELECT ng_internalid, ng_url, ng_tyng_language FROM ng_languagedependenturi WHERE ng_url = @url			5 715 757,11
	5	SELECT ng_internalid, ng_url, ng_type, ng_language FROM ng_languagedependenturi WHERE ng_url = @url			3 989 183,81
		ra. SELECT no internalid no url no tupe no language FBOM			0.000.047.70



10 NET Genium logging and log evaluation

- **1** The steps in the previous chapters should be used to identify the database in which the performance issues occur.
- Enable logging in the NET Genium settings by selecting "To database". If you are dealing with increased memory usage, also turn on "Measure memory consumption". A detailed description of NET Genia settings is given in a separate "NET Genium settings" manual.

General	Emails	Print templates	Permissions	Passwords	Colors	Other				
	E-Learning	● Style A ○ Style B								
	Logging Off									
		Logging to co Measure memory	nsole 🧿	n						
		Delete old log	is in the databas is on the disk	ie						
	Options	Enable autom	atic error report	ing to NetGeniu	ım					
		Enable sendin	ng lost password	s on the login p	age					
						Save				

- Evaluate logs with reports, and identify view pages, edit forms, database queries, scripts, or external functions that last the longest. A detailed description of the reports is given in a separate manual "Reports".
 - View pages (average_worker_time)
 - View pages (total_worker_time)
 - Edit forms (average_worker_time)
 - Edit forms (total_worker_time)
 - Database queries (average_worker_time)
 - Database queries (total_worker_time)
 - Scripts (average_worker_time)
 - Scripts (total_worker_time)
 - External functions (average_worker_time)
 - External functions (total_worker_time)



11 Editing SQL queries and program code

The most common cause of performance issues are improperly designed database queries or inefficient retrieval of data from the database. In most cases, you just need to set the indexes on the columns in the database correctly, or change the way the database tables are joined. It is far more laborious, but just as important, to change the way data is retrieved from the database so that as few queries as possible are sent to the database, and always run all queries of a given task with only one connection to the database.

11.1 Indexes

Indexes are used to speed up database queries, most often to speed up SELECT commands.

Where to set the index

• Enabling/disabling indexes for the "pid" and "pform" columns is performed on the "Administration" tab in the properties of the edit form.

General	Other	Help	Editing rights	Rights of chil	d objects	Administration						
	Identifie	er #sin	validlogins#		Col	rect Correct	all					
		Re Re Sele	ad/write data fron ad data from the ct command	table using using a vie	a memory optimize w	d table						
	Index 1 ● (None) ○ PID ○ PFORM + PID Index 2 ● (None) ○ (Custom)											
C# / Java	aScript	Statis	tics Loggir	ng		Save	Delete					



• Enabling/disabling indexes for all other database columns is performed on the "Administration" tab in the properties of the database control.

General	Default valu	ie Events	Help	Filter	Rights	Administration						
	Identifier	#date_# #sinvalidlogin	s#.#date	e_#		Correct						
Index Index column "Date" along with the following columns												
Other												
Curren the	nt value in database											
Chan	ge type to	TextBox 🗸										
Statistic	5					Sa	ave	Delete				

When and why to set an index

- When dealing with database performance optimization, it is important to follow the basic rule that every column that occurs in the join conditions on either the left or right side of the condition must be indexed. All other joins can be turned on/off only after some time of the application, when the database is gradually filled with data. Over time, the need to set up indexes changes, primarily according to the growing number of records in each database table, and consequently according to the way this data is read.
 - The (ID) column, which is the primary key of the database table, is often used on the left side of the join condition. This column is indexed automatically.
 - A foreign key the ForeignKey control is often used on the right side of the join condition. The index on this column is automatically turned on when the control is created, but it is possible to turn it off later.
- Gradual indexing on/off must be based on the basic nature of indexes indexing a given column only makes sense if the value of the column is evaluated using the operator "equals", "is defined" or "is not defined".
- A NET Genium logging to the database must be turned on before starting any index tuning.



Reports

- NET Genium includes reports with an analysis of all indexes in the database, along with a recommendation on where to turn on the index. In the case of the MSSQL database, it also contains reports with index fragmentation and index usage statistics. These reports are described in detail in the separate manual "Reports", chapter "Indexes".
 - In the "Joins" report, it is important to look for the occurrence of three "!!!" exclamation marks.
 Exclamation points indicate columns used in join conditions that do not have an index enabled. For these columns, it is important to turn on the index immediately.
 - All other entries in the "Joins" report are only informative and point to columns used in database query conditions for which the index may need to be turned on in the future.
 - The "Joins" report does not take into account queries that are used inside the source codes of external functions or console applications. Therefore, it only contains queries designed in the database query designer.

How indexes work

- ▲ Understanding the principle of indexes is key to properly designing indexes in a database. The following example in the language C# demonstrates searching for records in a database using a slow sequential traversal, followed by a fast dictionary search.
 - Sequential crawling is analogous to a situation where the database server does not have an index available and is therefore forced to crawl through all of the database records – one at a time – and try to find a match based on the search criteria.
 - Using a dictionary is analogous to a database server having an index on a given column and looking for all records that have one particular value stored in that column. For all other search criteria – other than matching a single specific value – you must use sequential traversal, and therefore cannot use an index.

```
using System.Data;
using System.Diagnostics;
// SELECT * FROM ng_table WHERE ng_tb IN ('10', '100', '1000', '10000', abc')
string[] search = new string[] { "10", "100", "1000", "10000", "abc" };
DataTable data = new DataTable();
data.Columns.Add("ng tb");
for (int i = 0; i < 5000000; i++)
{
  data.Rows.Add(i.ToString());
}
Stopwatch sw1 = Stopwatch.StartNew();
foreach (string s in search) Console.WriteLine(s + ": " + Find(data, s));
Console.WriteLine("Sequential browsing: " + sw1.Elapsed);
Console.WriteLine();
Dictionary<string, DataRow> dictionary = new Dictionary<string, DataRow>();
foreach (DataRow row in data.Rows)
  dictionary.Add(row["ng tb"].ToString(), row);
}
```



```
Stopwatch sw2 = Stopwatch.StartNew();
foreach (string s in search) Console.WriteLine(s + ": " + Find(dictionary, s));
Console.WriteLine("Dictionary: " + sw2.Elapsed);
Console.WriteLine();
Console.ReadLine();
foreach (DataRow row in dataTable data, string search)
{
  foreach (DataRow row in data.Rows) if (row["ng_tb"].ToString() == search) return true;
  return false;
}
private static bool Find(Dictionary<string, DataRow> dictionary, string search)
{
  if (dictionary.ContainsKey(search)) true;
  return false;
}
```

11.2 Joins LEFT JOIN

- SELECT * FROM ng_invoice LEFT JOIN ng_item ON ng_item.pid = ng_invoice.id
 - Load all invoices along with all items
 - In the resulting table, invoices are duplicated as many times as each invoice has items
 - In the resulting table there are also invoices that have no item (all columns from the joined item table have a NULL database value stored in the resulting table)
- SELECT * FROM ng_invoice LEFT JOIN ng_item ON ng_item.pid = ng_invoice.id WHERE ng_item.id IS NULL
 - o Retrieve all invoices that have no item
 - In the resulting table, there are only invoices that have no item (all columns from the joined item table have a NULL database value stored in the resulting table)
- SELECT * FROM ng_invoice LEFT JOIN ng_item ON ng_item.pid = ng_invoice.id WHERE ng_item.ng_price > 100
 - \circ Retrieve all invoices that have an item with a price greater than 100
 - In the resulting table, invoices are duplicated as many times as each invoice has items with a price greater than 100
 - This query is a classic example of "LEFT JOIN + Condition" where it is beneficial to change the slower LEFT JOIN to the faster INNER JOIN.



INNER JOIN

- SELECT * FROM ng_invoice INNER JOIN ng_item ON ng_item.pid = ng_invoice.id
 - o Retrieve all invoices that have at least one item
 - o In the resulting table, invoices are duplicated as many times as each invoice has items
 - o In the resulting table are only invoices that have at least no item

11.3 The most common errors in database queries

- SELECT * the query should never contain an asterisk as a wildcard for all columns of the source database table, as well as all columns of all accepted tables, but should always contain a comma-separated list of read columns.
- Loading strings of unlimited length (TextArea, RichTextBox, MultiListBox) causes a significant slowdown in database query processing, especially on the Firebird database server. These columns should only be loaded when absolutely necessary.
- Oetailed description of how to retrieve data from the database using C# including examples is given in the separate manual "External functions", chapter "Reading data from the database".

11.4 The most common errors when joining

- Failure to follow the basic rule that every column that occurs in join conditions on either the left or right side of the condition must be indexed.
- Joins defined in the database query designer are often either completely useless because the result does not use columns retrieved from the accepted database table, uses a slower "LEFT" instead of a faster "INNER" in cases where "LEFT" is unnecessary, or uses "LEFT" in conjunction with the database condition. query, see the "Reports" manual, chapter "Database Queries (LEFT JOIN + Condition)".
- The moment large database tables join together, or join them too much, you need to completely change the data retrieval logic, and split a costly database query with many joins into more simple database queries.



11.5 SQL Server Database Engine Tuning Advisor – SQL query debugging

SQL Server includes a useful tool for debugging database queries – SQL Server Database Engine Tuning Advisor. A simplified procedure for debugging database queries is provided in the "Config\Tools\TunningQueries.txt" every NET Genium:

1) Run SQL Server Management Studio 2) Right click on the instance name and select 'Reports / Standard Reports / Performance - Top Queries by Total CPU Time' 3) Identify top queries with constants in a condition that can be improved with indexes 4) Right click on the background of the report and select 'Print / Excel' 5) Open printed Excel file 6) Create 'netgenium.sql' file and insert top queries using the following syntax: use netgenium go SELECT ... SELECT ... 7) Run SQL Server Database Engine Tuning Advisor 8) Click 'Start New Session' 9) Select 'File' as a 'Workload' and browse for 'netgenium.sql' 10) Mark 'netgenium' database 11) Click 'Start Analysis' 12) Analyze 'Recommendations' tab 13) Create new indexes as recommended