SQL Server Whitepaper

SQL SERVER FRAGMENTATION EXPLAINED

BY JUAN ROGERS

Learn about fragmentation and how to improve performance with SQL Defrag Manager

INTRODUCTION

As the data in Microsoft SQL Server tables changes their indexes change. Over time these indexes become fragmented. This fragmentation will adversely affect performance. This technical white paper provides information to help you understand the detailed mechanics behind fragmentation. It will also help you understand the methods and approaches for performing defragmentation so you can improve your SQL Server's performance.

This technical white paper will help you understand SQL Server fragmentation and the performance benefits you can gain on your SQL Servers by continuously monitoring and managing index fragmentation.

The following is a summary of the key topics covered in this paper:

- The difference between disk and SQL Server internal and external fragmentation
- How fragmentation affects performance
- The mechanics behind performance robbing data voids
- The pros and cons of various approaches to managing fragmentation
- How SQL Defrag Manager provides a better, more efficient and automated approach to identifying and resolving index fragmentation in SQL Server
- How to judge the improvements gained by defragmenting your server

Warning: This white paper will get a bit technical as it is intended for DBAs who want to truly understand the details and key components of fragmentation in SQL Server.

Q WHAT IS SQL SERVER FRAGMENTATION? IS IT DIFFERENT THAN PHYSICAL DISK FRAGMENTATION?

A SQL FRAGMENTATION IS NOT PHYSICAL DISK FRAGMENTATION. NOT ALL FRAGMENTATION IS EQUAL!

Physical disk fragmentation is likely what comes to mind when fragmentation is first discussed. Physical fragmentation is a side effect of how hard drives and Windows work. It is common knowledge that regular disk defragmentation is required to achieve optimal performance from your PC. Windows even includes a basic defragmentation utility.

Physical fragmentation slows down your PC because reading data is interrupted by head seek delay. Windows fits files into free space, often breaking the file into segments stored apart from one another. A hard drive's head relocates to read each individual segment. As it moves to each segment the head 'seeks' - often at a cost of 3-4 times the time it takes to read the segment itself. Physical fragmentation primarily affects desktop or laptop PCs containing one hard drive. The single drive must sequentially gather data – so on a fragmented disk it seeks, reads, seeks, reads - these 4 operations are performed one after another. Defragmented, the operation ends up as seek, read, read. We reduce the total cost of 24ms to 15ms in our simple example.

Physical defragmentation products such as Windows defrag, Power Defrag™, Page Defrag™ (another Microsoft tool), or the granddaddy of them all, Diskeeper 2011™ work very well when repairing segmented files. Diskeeper's technology is licensed to Microsoft as the defragmentation tool internal to Windows. In fact, Diskeeper's latest innovations bring physical defragmentation capabilities to a completely new level. All of these products reorder the data on your disk, consolidating files into fewer segments to minimize "head seeks" – providing faster boot times, quicker file reads, and a more responsive system overall.

However, physical disk fragmentation is not the same as SQL Server defragmentation! SQL Server is different. SQL Servers use advanced storage systems with multiple drives working in tandem, changing the way files are read. Physical fragmentation is something solved with hardware – not with defragmentation scripts or tools.

The fault-tolerance in database storage overcomes the vast majority of physical disk fragmentations' impact. Best practices universally prescribe multi-drive storage subsystems for production SQL Servers. Most SQL Servers use multi-drive storage such as RAID arrays, SANs, and NAS devices; there are always multiple drives acting in tandem. Hard disk controllers supporting drive arrays are aware of the alternate seek/read dynamic and tailor communications with the array for maximum I/O.

As a result, files are distributed across many drives inherently becoming segmented. Working in tandem, however, allows one drive to seek while the others read. With the common configuration of 5 drives, a seek delay of 9ms per drive allows 2 drives reading for 3ms with no seek delay impact at all. Data storage drives are generally much faster than workstation drives, so seek times of 4ms and read times of 1.5ms are not unusual.

There are many DBAs who run a traditional physical defragmentation program in tandem with their intelligent drive controller which results in limited improvement. Physically defragmenting a file in an array implicitly leaves the file segmented across the virtual unison of tandem drives. It's by design. The goal is to gain the most performance while incurring the least overhead – so don't run physical defrags if they slow the storage by 50% while running, and ultimately improve read speeds 1-2%.

The most important concept to understand is that the controller, physical defragmentation programs, and multi-drive arrays are unaware of what SQL Server is doing with the file data internally. By focusing on SQL Server's representation of data - how SQL Server has laid out the database itself, how full each page is, and how effectively we're utilizing available SQL Server resources, we can optimize to the 'next level' of SQL Server performance, solidly trumping any benefit to physical defragmentation by orders of magnitude. In a nutshell, SQL Server's performance can be most improved by focusing on its internals. In fact, once you start focusing on defragmentation at the SQL Server level – whether with manual defragmentation or with the automated defragmentation provided with SQL Defrag Manager, you may decide that physical defragmentation is no longer needed!

Q HOW IS SQL SERVER'S FRAGMENTATION AFFECTING MY SERVER?

A FRAGMENTATION MAINLY CREATES WASTED SPACE THAT CAN AFFECT SERVER PERFORMANCE MORE THAN YOU'D EXPECT.

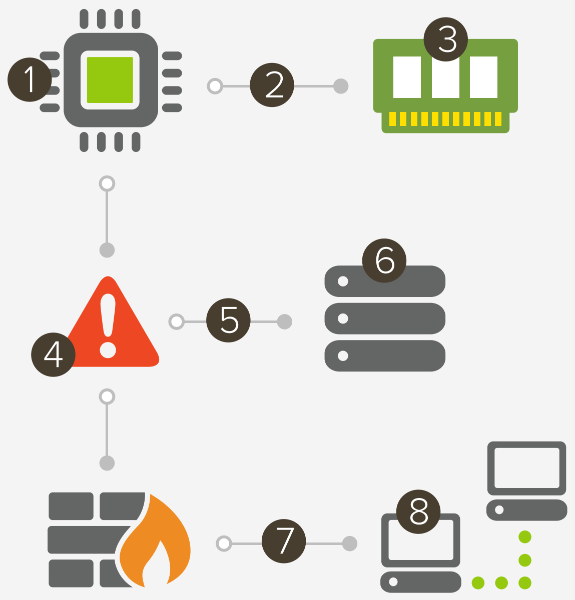
Fragmentation of your SQL Server's internal allocations and page structures result in 'gaps' or 'void' space that is dead weight carried along with valid data. Your backups, storage, I/O channels, buffer memory, cached data, logs, tempdb, CPUs and query plans are impacted by these unnecessary voids. SQL's fragmentation continually eats away at these resources with nearly every select, update, delete, insert, and table/index change. If ignored, fragmentation can be the proverbial 'death by a thousand cuts' to a server's performance and scalability.

Q WHAT CREATES THE VOIDS AND OTHER ADVERSE EFFECTS AND HOW DO I GET A HANDLE ON THEM?

A TYPICAL, DAY TO DAY ACTIVITY CAUSES SQL SERVERS TO FRAGMENT OVER TIME.

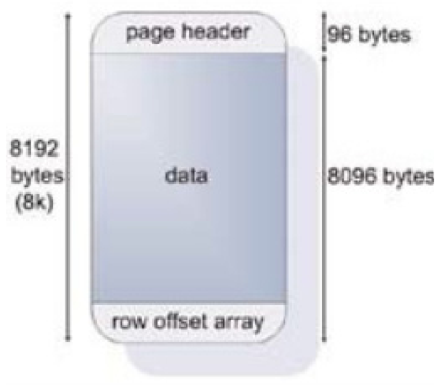
Changes to your data – inserts, updates, deletes, and even changing varchar values contribute to fragmentation. The full list of actions that cause fragmentation is long and the rate of fragmentation varies across different indexes and tables. Sometimes there is a pattern resulting from seasonal or annual peaks (e.g., when financials run). But more often than not, it is difficult to find, predict and proactively manage fragmentation manually.

Shown here is a diagram of how SQL Server fragmentation can affect your SQL Server performance and overview of the affected areas. As you identify how the fragmentation affects your server, you'll see that fragmentation effects are cumulative and nearly impossible to predict. SQL Defrag Manager, however, uses sophisticated algorithms to predict and detect SQL Server fragmentation "hot spots" and to defragment indexes on a continuous basis.



1. The CPUs cannot move data in nor out as efficiently as possible. They also operate less effectively internally as their internal comparison registers are comparing superfluous void data. They cannot symetrically multiprocess as effectively as concrete synchronization points are fewer.
2. RAM to CPU I/O is affected, hampering both the current request and any concurrent requests.
3. The server's memory, while still storing 4GB, has less data to work with due to the void space. Causes greater I/O for writes, more disk activity because of page flushes, and less efficient queries.
4. The most likey physical weak links are designated by this bottleneck. We're fortunate if this is what first illustrates our problem, as they are easy to fix compared to digging to find where we've tipped the scale.
5. The storage I/O channels are subject to the same inefficiences we see elsewhere. The strategizing controllers are less effective, and there is significantly more traffic between storage and server than is necessary.
6. Our storage devices are saddled with storing void data, so the per-drive, per-LUN, and per-controller caches waste valuable space. Strategizing controllers aren't able to do well with void space.
7. Painfully obvious, every effect caused by fragmentation – from the smallest ripple to the largest catastrophe is felt by the users. Often it's a slow, gradual decline in performance, most obvious when the application is in production and no development has been taking place. It grinds more and more slowly until SQL decides to table scne instead of use an idex... Ouch!
8. Also obvious, but the total number of clients that can be supported is directly dependant on how well you're managing your server's use of its resources. How do you know? (Waiting for the help desk call isn't a valid answer)

SQL Server stores all data, objects, and internal structures in 8192 byte data pages shown in Figure 2. These pages are known only to SQL Server and may be stored in one or more physical files on disk. Data gets a maximum of 8096 bytes per page – the rest of the page contains the page header and row locations. When creating a table or index, SQL Server pages fill according to the fill factor you specify (or the closest approximation.)



Over time, insert, deletes, and modifications (such as widening the value in varchar fields) fill the page and ultimately overflow the page creating a 'page split'. Splitting divides the full page evenly, putting half of its data on a newly allocated page, and may negate any fill factors you designate. For example, if you designate a fill factor of 80%, over time, due to splitting , your pages may reach a fill factor of 50% or less.

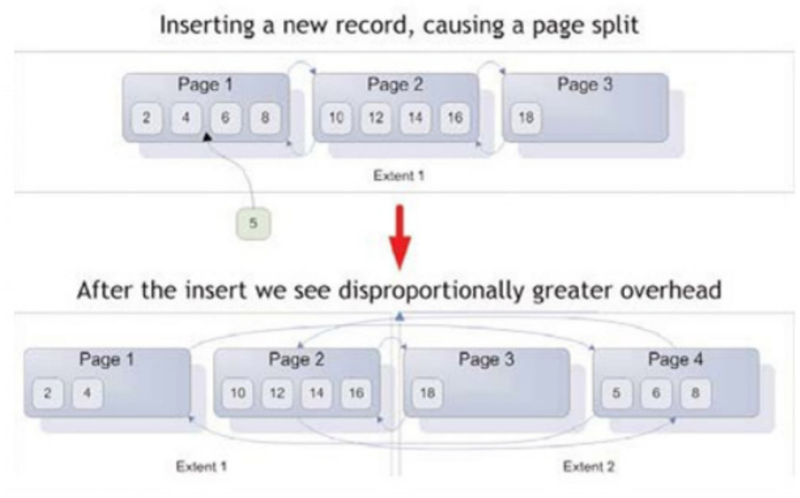
The more that heavily, spiked, or continuous changes occur on a table, the faster and further it and its indexes drift. Since the indexes are based on variants of data in the table, they have their own unique drift profile.

The net result of drifts is waste – lots of it – waste of your disk, I/O channels, server's caches and buffers, and CPU utilization. The waste may also skew your query plans.

The void/waste space is known as "internal fragmentation." Internal fragmentation lowers page density and as a result our server resources trickle slowly away now being increasingly consumed by empty space. SQL does try to fill the voids in split pages – however there is rarely the Tetris™-like fit necessary to reach optimal population post split. The common practice of using an identity column as your clustered index, forces inserts into new pages at the bottom of the table, preventing recovery of the voided space.

The space used by actual data is reflected in a metric called "page density." The denser a page, the more data vs. void it contains. A page density of 100% would mean the data page is completely full. Even if the pages had no void, Figure 3 illustrates how the split has introduced other inefficiency in contiguously accessing the pages after the split. Interestingly this parallels physical fragmentation – although it is a completely isolated variant in SQL Server's management of data vs. the way files are segmented on disk. This type of fragmentation is called 'external fragmentation'.

The vast majority of the time, the void grows, instead of filling up neatly. When void space becomes too much (your page density becomes too low), SQL Server will discard the index due to excessive overhead. At this point, fragmentation becomes very evident as very few systems will tolerate discarding indexes in favor of table scans.



While it may seem trivial on a small scale, when your average page density is low, you are wasting disk space, incurring more physical I/O, increased logical reads, wasting precious server memory while computing and comparing data unnecessarily. Further, if you are fortunate enough to have an intelligent I/O controller, you are also wasting the benefit of its optimization strategies. It becomes obvious that this process of splitting, voids, progressive order, and rates of decay requires non-stop attention to insure the server is running with as much free resource as it can.

Q OK, SO CLEARLY THE MAINTENANCE BENEFITS ARE REAL, AND WE WANT TO OBTAIN THEM, NOW WHAT?

A BESIDES **SQL DEFRAG MANAGER** FROM IDERA, THE TWO MOST COMMON APPROACHES FOR FRAGMENTATION BOTH HAVE SERIOUS DISADVANTAGES.

If, for a moment, we ignore SQL Defrag Manager, there are two existing methods for managing SQL Server fragmentation. Neither is ideal, or gives you the information you need to stay informed and on top of the fragmentation challenge. Both leave you completely blind — you won't know if they helped, hurt, stepped on, or blocked your busiest table.

METHOD 1 REACTIVE DAMAGE CONTROL

The server performance degrades slowly and is ignored. All of a sudden, a spot in the database reaches critical mass, performance craters, and is eventually addressed. This is how the majority of DBAs are introduced to the fragmentation problem. They just fix it and wait for the next hotspot or for SQL Server performance to run down again and again. Unfortunately, you will never know when your server is going to act up or how severe the impact will be. Furthermore, there may be cascade effects caused by inadvertent query plan disruption due to fragmentation.

METHOD 2 RUN A BLIND MAINTENANCE SCRIPT

These scripts are often quite complex with unpredictable results. They usually work, but may often cause after effects such as blocking or locking and can generate considerable overhead. You have no idea how long it will take the script to find every index – and it must query each one, every time. The script does not track performance benefits, does not track how defragmentation varies each time, and offers no notification of potential problems.

All-purpose SQL Server defragmentation scripts:

- Request information that can cause contention or deadlocks.
- Rarely have internal logic to know when to defragment– instead they just steamroll your servers every day whether they need it or not (perhaps many times a day.)
- May require changes which would required you to re-deploy the new script to all of the servers in your enterprise.
- Should be tailored to each database – but to do this would require near constant "hand-tuning." A very time-consuming and practically impossible process.
- Aren't able to report when the script was run, what performance enhancements were gained, or how many resources they've reclaimed on your server since you started running them.

METHOD 3 (THE BEST ONE)
SQL DEFRAG MANAGER

SQL Defrag Manager offers a totally new way to identify, optimize, manage and automate SQL Server defragmentation. It is designed specifically to overcome the compromises DBAs have to make regarding the important task of index fragmentation maintenance.

Consider this: If you are able to eliminate void space, every page of void reclaimed is money back in your corporation's pocket. Those reclaimed resources are regained server capacity that had been lost unnecessarily. SQL Defrag Manager will reclaim these resources and track the total improvement on every object in your enterprise daily or over a year. You can even produce an annual report showing how much money has been saved through the use of defragmentation technology – and we guarantee that it will be impressive!

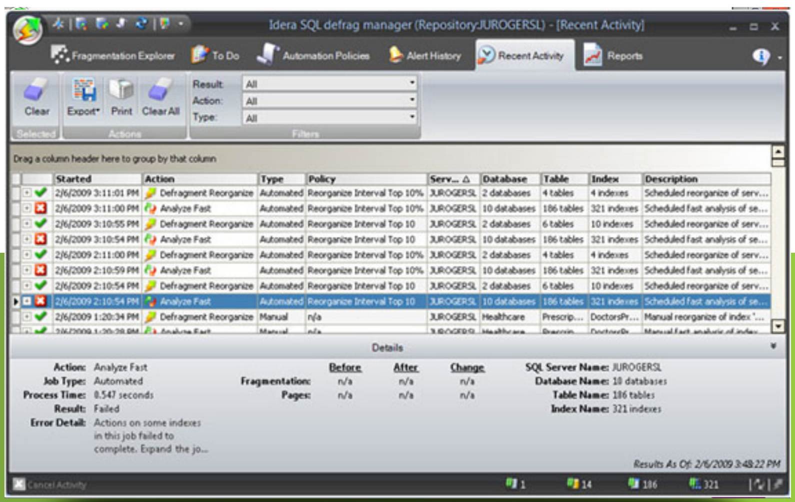
Futhermore, SQL Defrag Manager brings proactive intelligence to the scheduled defragmentation job. Specifically, it gives the DBA the ability to ascertain the status of system resource metrices prior to executing the defrag policy. The DBA can set thresholds for (Server CPU %, SQL server CPU %, Memory,TLOG % full) and much more!

The "Proactive Resource Check" makes it possible for the DBA to proactively anticipate unplanned outages or system bottle-necks that may cause application batch cycles to creep into the defragmentation maintenance window, which may prevent it from running. If the resource check exceeds the customer defined threshold, a notification is sent to the DBA.

SQL Defrag Manager not only tracks the improvement achieved on each object, it maintains dozens of statistics on each table and index. This information guides SQL Defrag Manager to determine how often it should check for fragmentation, and if you wish, the method it will use to correct the fragmentation. SQL Defrag Manager eliminates defragmentation overhead and risk on your servers – there is no agent required on any managed server. There is no job scheduled or script deployed. SQL Defrag Manager simply runs as a service, quietly in the background with no affect to your production servers.

Unlike scripts, SQL Defrag Manager's fragmentation detection routines are nonblocking. Defragmentation is also non-blocking, given the DBA has not chosen to rebuild the fragmented object. Rebuilt objects are often not needed. SQL defrag was developed based on the feedback from experienced DBAs who were frustrated with the scripts and the handholding that their 24x7 , 99.999% available enterprises required. SQL Defrag Manager will shed light on the fragmentation levels across your entire SQL Server environment — allowing you to quickly detect and manage fragmentation with ease. It will also give you assurance that defragmentation is being handled in exactly the way it should be for that particular database – no more guessing!

You be the judge of how we've done.



Result: All
Action: All
Type: All

Drag a column header here to group by that column

	Started	Action	Type	Policy	Serv... △	Database	Table	Index	Description
✓	2/6/2009 3:11:01 PM	Defragment Reorganize	Automated	Reorganize Interval Top 10%	JUROGERSL	2 databases	4 tables	4 indexes	Scheduled reorganize of serv...
✖	2/6/2009 3:11:00 PM	Analyze Fast	Automated	Reorganize Interval Top 10%	JUROGERSL	10 databases	186 tables	321 indexes	Scheduled fast analysis of se...
✓	2/6/2009 3:10:55 PM	Defragment Reorganize	Automated	Reorganize Interval Top 10	JUROGERSL	2 databases	6 tables	10 indexes	Scheduled reorganize of serv...
✖	2/6/2009 3:10:54 PM	Analyze Fast	Automated	Reorganize Interval Top 10	JUROGERSL	10 databases	186 tables	321 indexes	Scheduled fast analysis of se...
✓	2/6/2009 2:11:00 PM	Defragment Reorganize	Automated	Reorganize Interval Top 10%	JUROGERSL	2 databases	4 tables	4 indexes	Scheduled reorganize of serv...
✖	2/6/2009 2:10:59 PM	Analyze Fast	Automated	Reorganize Interval Top 10%	JUROGERSL	10 databases	186 tables	321 indexes	Scheduled fast analysis of se...
✓	2/6/2009 2:10:54 PM	Defragment Reorganize	Automated	Reorganize Interval Top 10	JUROGERSL	2 databases	6 tables	10 indexes	Scheduled reorganize of serv...
✖	2/6/2009 2:10:54 PM	Analyze Fast	Automated	Reorganize Interval Top 10	JUROGERSL	10 databases	186 tables	321 indexes	Scheduled fast analysis of se...
✓	2/6/2009 1:20:34 PM	Defragment Reorganize	Manual	n/a	JUROGERSL	Healthcare	Prescrip...	DoctorsPr...	Manual reorganize of index '...
✓	2/6/2009 1:20:28 PM	Analyze Fast	Manual	n/a	JUROGERSL	Healthcare	Prescrip...	DoctorsPr...	Manual fast analysis of index

Details

Action: Analyze Fast
Job Type: Automated
Process Time: 0.547 seconds
Result: Failed
Error Detail: Actions on some indexes in this job failed to complete. Expand the jo...

	Before	After	Change
Fragmentation:	n/a	n/a	n/a
Pages:	n/a	n/a	n/a

SQL Server Name: JUROGERSL
Database Name: 10 databases
Table Name: 186 tables
Index Name: 321 indexes

Results As Of: 2/6/2009 3:48:22 PM

1 14 186 321

SQL DEFRAG MANAGER

Within moments of installation you'll see a screen like that to the right. Fragmentation level detection in SQL Defrag Manager is non-blocking. Unlike scripts or manual queries, risk of unpredictable impact is eliminated. You have immediate and complete visibility into your enterprise and how fragmentation is affecting it.

Displaying all the servers in your enterprise, each is color coded to reflect how much impact fragmentation is having on it, its databases, tables and indexes. You can easily sort to bring the items most in need of your attention to the forefront.

Quickly drill down into the server, database, and locate an offending table/index by clicking the pie chart. If you want to fix it, one click safely fixes it (again, using non-blocking techniques) and reflects the improvement right away. Once you find the fragmented spots, let SQL Defrag Manager to make sure it never becomes a problem again. You can set it to send you a summary and let you know if there is a problem, or it can simply defragment as needed – automatically.

INTELLIGENT SCHEDULING

Setting the schedule you wish for the object to be checked is straightforward. Start and end ranges can be specified to insure that all activity is performed to your instruction and can be restricted to predefined lull or 'safe' windows. You can also have it run a one time job at a convenient time for you — such as on the weekend.

IS IT SAFE TO DEFRAGMENT?

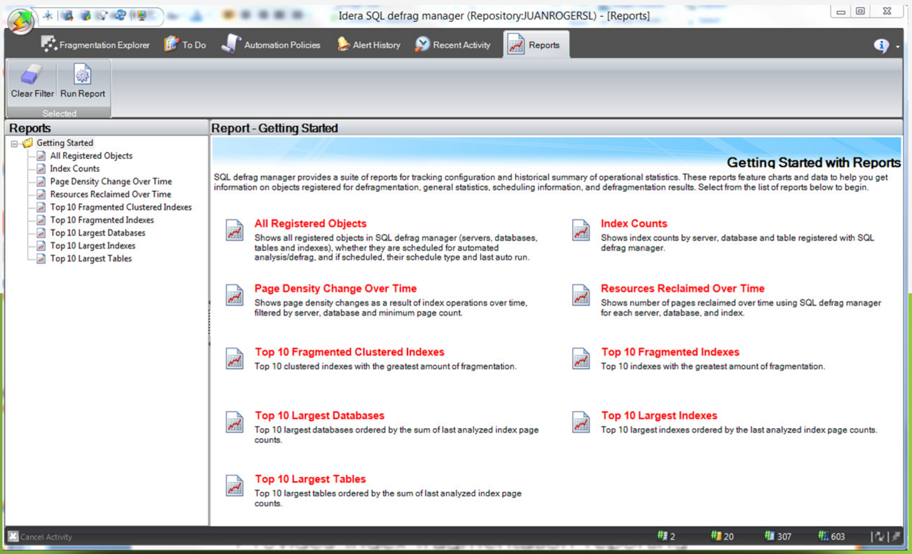
The DBA can define thresholds for 9 system resource metrics prior to executing the defrag policy.

FREQUENT INDEX DEFRAGMENTATION? MANAGE INDEX FILL FACTOR

If your indexes are getting too fragmented, you can reduce your defragmentation by modifying the index fill factor.

DEFRAGMENT YOUR WAY - WHILE INSURING MAXIMUM BENEFIT.

- Have it notify you of approaching thresholds, so you can manually defragment when you wish.
- Have it try non-blocking remediation and if that cannot reach an acceptable level, tell you.
- Have it try non-blocking remediation, and if that doesn't reach acceptable, try more invasive blocking.



Report - Getting Started

Getting Started with Reports

SQL defrag manager provides a suite of reports for tracking configuration and historical summary of operational statistics. These reports feature charts and data to help you get information on objects registered for defragmentation, general statistics, scheduling information, and defragmentation results. Select from the list of reports below to begin.

All Registered Objects
Shows all registered objects in SQL defrag manager (servers, databases, tables and indexes), whether they are scheduled for automated analysis/defrag, and if scheduled, their schedule type and last auto run.

Index Counts
Shows index counts by server, database and table registered with SQL defrag manager.

Page Density Change Over Time
Shows page density changes as a result of index operations over time, filtered by server, database and minimum page count.

Resources Reclaimed Over Time
Shows number of pages reclaimed over time using SQL defrag manager for each server, database, and index.

Top 10 Fragmented Clustered Indexes
Top 10 clustered indexes with the greatest amount of fragmentation.

Top 10 Fragmented Indexes
Top 10 indexes with the greatest amount of fragmentation.

Top 10 Largest Databases
Top 10 largest databases ordered by the sum of last analyzed index page counts.

Top 10 Largest Indexes
Top 10 largest indexes ordered by the last analyzed index page counts.

Top 10 Largest Tables
Top 10 largest tables ordered by the sum of last analyzed index page counts.

CUSTOM TAILOR DEFRAGMENTATION TO YOUR ENTERPRISE – SCHEDULE, SET THE RANGE, AND SET THE ACTION.

- If you set only the server level all databases, tables, and indexes will inherit.
- Override any table you feel should be handled differently – perhaps with a custom range or time.
- Limit activity to outside specific windows on particular days – insure balanced operation even on weekends.
- Defragmentation running too frequently? Alter index “fillfactor” and see the benefits over time.

LET SQL DEFRAG MANAGER DEFRAGMENT FOR YOU.

- The tool is as interactive or automated as you wish it to be.
- There is no need for you to watch every potential problem in your enterprise. SQL defrag will let you know.
- SQL Defrag Manager brings proactive resource checking intelligence to the defragmentation process

CENTRAL ENTERPRISE MANAGEMENT CONSOLE – RUN THE CLIENT FROM ANY MACHINE.

- The SQL Defrag Manager Management Console provides a real-time window into fragmentation levels.
- Sit at your desk - view and manage defragmentation activity across all of your servers.
- Agent-less collection mechanism. There are no resident scripts on any of your monitored servers.

LIGHTWEIGHT COLLECTION

- Fragmentation details are intelligently collected based on a customizable schedule, keeping overhead on your monitored servers low and controlled.
- If SQL Defrag Manager predicts a problem will occur before it is next scheduled, it’ll let you know.

10,000 FT. TO 1 FT. + 360° REPORTING -

- Comprehensive overview reporting covers every aspect of your enterprise at a glance.
- Detailed breakdown reports show exactly how each object has been managed, and the improvement.
- Summary improvement – see how much SQL defrag has benefitted you and your company – every day.
- All of the reports are in Reporting Services. Subscribe to your reports and read with your morning coffee.
- Model-based reporting allows for easy development of your own custom reports.

RESOURCES RECLAIMED REPORTS: PERHAPS THE MOST IMPORTANT REPORT OF ALL?

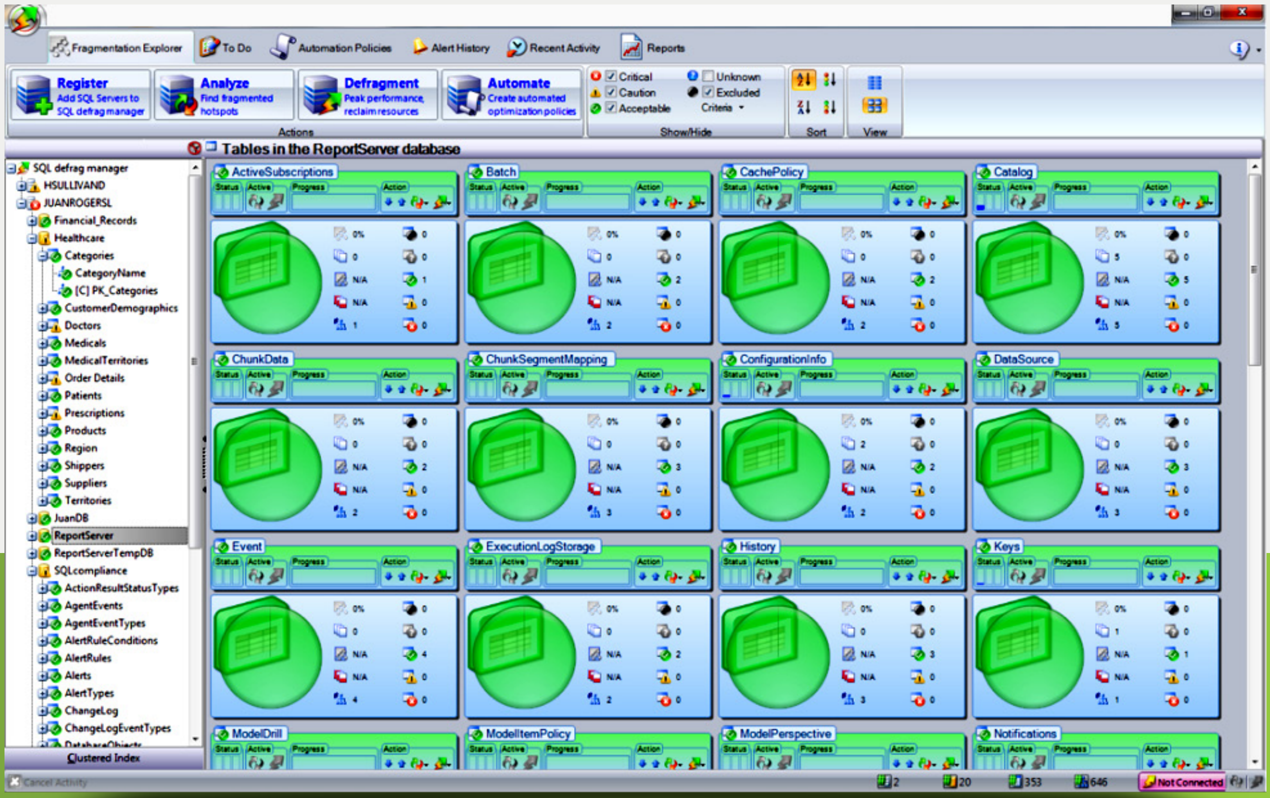
SQL Defrag Manager reports on all resources reclaimed during defragmentation and will translate this into memory, disk, I/O, CPU, and backups – and assign hard costs to each. You assign costs specific to your environment so that you have a concrete ROI value for defragmentation that you can provide management. Perhaps if you save your company $50,000 they might consider that $10,000 raise?

SQL DEFRAG MANAGER

FIND AND FIX SQL FRAGMENTATION

- Automate identification of index fragmentation “hot spots”
- Schedule index defragmentation jobs
- Avoid database contention with system resource pre-checks
- Reduce page splits with index fill factor settings control
- Centrally manage, report and notify instantly

Start for FREE



IDERA

IDERA.com

IDERA understands that IT doesn’t run on the network – it runs on the data and databases that power your business. That’s why we design our products with the database as the nucleus of your IT universe.

Our database lifecycle management solutions allow database and IT professionals to design, monitor and manage data systems with complete confidence, whether in the cloud or on-premises.

We offer a diverse portfolio of free tools and educational resources to help you do more with less while giving you the knowledge to deliver even more than you did yesterday.

Whatever your need, IDERA has a solution.

IDERA