Environment Optimization Guide

November 24, 2021 | Version Server 2021

Table of Contents

1 Overview

This guide outlines the best practices for maintaining and optimizing a Relativity environment. Follow these recommendations to ensure stability and optimal performance of all workspaces.

1.1 Release updates

Note the changes and additions to this guide for each service release:

- We typically release a new major version of Relativity once a year.

- Product updates are released once a month. You can view Relativity release notes on the Documentation site.

- Verify that you're using the latest version of this guide. We update this guide as new information is obtained from field deployments.

1.2 Training and support

This guide is often provided to system admins who may not have any exposure to the Relativity interface. For more information, you can access the following resources:

- **User tutorials** - our Relativity tutorials are designed to help users become familiar with Relativity quickly. Using interactive tutorials that cover specific Relativity features, you can experience hands-on learning at your own pace. To view these tutorials, visit the Tutorial page on relativity.com.

- **Documentation** - we constantly strive to ensure that all users of Relativity are educated on the full functionality of the platform. To help you and your team stay well informed, we post all documentation on this website. To access documentation from previous versions of Relativity, see the documentation archives page.

- **Relativity Community** - you can access the Relativity Community from the Relativity.com Support page. If you need a Relativity Community account, contact support@relativity.com.

1.3 Environment optimization checklist

Use the following checklist to help manage and plan for your Relativity environment.

Reference	Task	Done
Release updates above	Review patch releases once a month.	
Training and support above	Review user tutorials.	
Training and support above	Review Relativity documentation.	
Training and support above	Explore the Relativity Community.	
Configuring Windows server on page 11	Configure the Windows server for optimum performance.	

Reference	Task	Done
Microsoft Windows server service packs on page 11	Verify that the latest Microsoft server service pack is installed.	
Windows power plans and CPU performance on page 11	Configure Windows power plans for high performance.	
Windows visual effects on page 12	Configure Windows visual effects for high performance.	
Windows processor scheduling on page 12	Configure Windows processor scheduling.	
Windows NTFS fragmentation on page 13	Install an automatic disk defragmentation tool.	
Configuring virtual memory on page 13	Manually set the size of the paging file to 4095 MB or higher.	
Antivirus directory exclusion list on page 14	Exclude the SQL Server, agent server, web server, Analytics indices, dtSearch indices, and the file repository from the antivirus software.	
Management server on page 16	Create a management server with SSMS and the RDC.	
Configuring SQL Server on page 17	Configure SQL Server for optimum performance.	
Optimize for ad hoc workloads option on page 17	Optimize SQL Server for ad hoc workloads.	
Max degree of parallelism option on page 17	Set max degrees of parallelism.	
Optimize the TempDB database on page 20	Optimize the tempdb databases.	
SQL maximum server memory on page 20	Set SQL Server max memory.	
Instant file initialization on page 22	Enable instant file initialization.	
File allocation unit size on page 23	Set the file allocation unit size.	
Microsoft SQL Server service packs on page 23	Verify that the latest SQL Server service pack is installed.	
Autogrowth settings on page 24	Set auto growth settings for SQL Server files.	

Reference	Task	Done
SQL Server best practices analyzer on page 24	Install and run SQL Server 2012 Best Practices Analyzer.	
RAID levels and SQL Server on page 25	Review RAID Levels and SQL Server storage options: ■ Verify that the tempdb data files reside on the fastest disks. ■ Verify that data, log, full-text, and tempdb files reside on separate disk volumes.	
Java heap size (JVM) on page 30	Set the Java heap size to one-third of total RAM.	
Configuring the Analytics server on page 29	Configure the Analytics temporary directory.	
Index directory requirements on page 31	Store the Analytics index locally, if possible.	
Setting up your workspace on page 36	Set up you workspace following best practices.	
Fixed-length vs. long text field types on page 36	Use fixed-length fields when possible.	
Fixed-length text field considerations on page 36	Set your fixed-length text fields to the appropriate size.	
Unicode support on page 37	Define the Unicode Enabled field property prior to import.	
Data imports on page 37	Don't run the Relativity Desktop Client on a Relativity production server.	
Views and searching on page 38	Follow best practices for views and searching.	
Tally/Sum/Average mass operation on page 38	Create indexes on groups that you Tally on.	
Group by for Pivot on page 39	Create a non-clustered index on fields that are being grouped on for Pivot.	
User notifications on page 39	Learn how to use various user notification methods.	
Analytics performance considerations on page 42	Review Analytics performance considerations.	
Monitoring environment performance on page 40	Ensure you have the right amount of memory when creating indexes.	

Reference	Task	Done
Monitoring environment performance on page 40	Review native imaging and processing performance considerations.	
Managing your Relativity environment on page 46	Review maintenance plan recommendations.	
SQL backups on page 48	Create backups for: ■ Relativity system databases ■ Relativity system database logs ■ System databases and logs	
Full backups on page 49	Ensure Verify backup integrity is selected for full backups.	
SQL recovery models on page 50	Set the SQL recovery models.	
Relativity data backups on page 50	Create backups for: ■ dtSearch and Analytics index shares ■ Relativity web server install directories ■ Native and image file shares	
Check Database Integrity task on page 50	Schedule a weekly check database integrity test.	
SQL table index management on page 51	Set up and schedule the IndexOptimize smart script for index and statistics maintenance.	
Updating statistics on page 51	Update statistics to improve query performance.	
Database log management on page 51	Review database log management recommendations.	
Shrink Database task on page 52	Ensure AUTO_SHRINK is not enabled.	
Job email notification alerts on page 53	Set SQL Database Mail on all Relativity SQL Servers and add these notification alerts to all scheduled Relativity maintenance tasks.	
Monitoring environment performance on page 40	Review environment monitoring guidelines.	
Windows and SQL Server log analysis on page 40	Set up these alerts for SQL and Windows server logs: ■ Low disk space on the servers ■ Server becomes unresponsive	

Reference	Task	Done
	■ Website becomes unavailable	
Resource utilization on page 40	Gather benchmarks for Relativity servers' resources.	
Monitoring disk usage on page 40	Measure disk latency.	
Workspace management and maintenance on page 46	Create needed indexes on workspaces with 500,000+ records.	
Analysis of long running queries on page 47	Analyze and try to optimize long-running queries.	
Full-text index management on page 47	Manage full-text indexes.	
Audit record table on page 48	Manage the Audit Record Table, if needed.	
SQL Server table index fragmentation on page 44	Understand, identify, and remove fragmentation.	
Expanding your Relativity environment on page 54	Review Expanding your Relativity environment.	
Agents on page 54	Scale agents as needed.	
Relativity Desktop Client import or export on page 55	Create a dedicated web server for the RDC.	
Expanding your Relativity environment on page 54	Create a dedicated agent server for the dtSearch query requests.	
Physical memory on the SQL Server(s) on page 55	Increase RAM on the SQL Server if needed.	
SQL Server failover clustering on page 56	Add additional servers for increased throughput and performance as needed.	
Distributed Relativity SQL Servers on page 56	Create a distributed SQL Server if needed.	
Web load balancing on page 57	Use a web load balancer if needed.	
Terminal services on page 58	Create terminal servers if needed.	
Bandwidth tester on page 58	Test the capacity of a network connection with the Relativity bandwidth tester.	

2 Configuring Windows server

Use the following guidelines to configure your Windows Server for optimum performance with Relativity. Some of these configuration options are one-time settings, while others require intermittent updating as your hardware or database sizes change.

> **Note:** These guidelines are applicable to all SQL Servers in the environment, including the Worker Manager SQL Server and the workers.

2.1 Microsoft Windows server service packs

Install the latest Microsoft Windows Server Service Pack on all Relativity servers.

However, compatibility for higher .NET versions is not guaranteed and we do not recommend installing higher .NET versions than what is listed as required by your Relativity version. Furthermore, install any smaller security patches, Windows updates, etc. at your own discretion. We only test major service packs, not every Microsoft update. Deploy any patches to your test instance of Relativity first. Ensure that a rollback plan is in place if you discover any issues during deployment.

Ensure you disable the option to Install updates automatically on all Relativity servers. Apply any required updates during a planned maintenance window.

2.2 Windows power plans and CPU performance

In Windows Server 2012 R2 and 2008 R2, you can access the Power Options screen by entering **powercfg.cpl** in a Run prompt. Three default power plans are available: Balanced, Power saver, and High performance. Balanced is set by default, which is what Microsoft recommends for general use. Unfortunately, the default Balanced power plan may not be the best choice for production server performance.
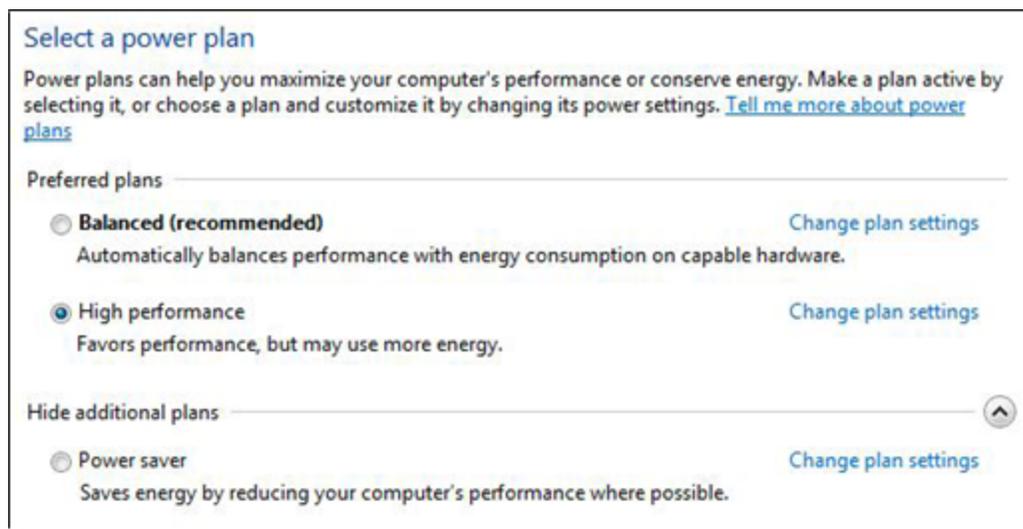
We recommend configuring all Relativity SQL Servers to use the High-performance windows power plan. This may also require changes to BIOS settings. You must decide what is best to accommodate all Relativity server roles.

2.3 Windows visual effects

Windows includes standard visual effects to make the user experience more enjoyable. These effects aren't required and consume CPU resources. We recommend disabling these effects on all Relativity servers.

1. Select the Performance Options dialog by clicking **Settings** in the Performance panel.

2. Select the **Visual Effects** tab in the Performance Options dialog.

3. Select the **Adjust for best performance** option, and then click **OK**.
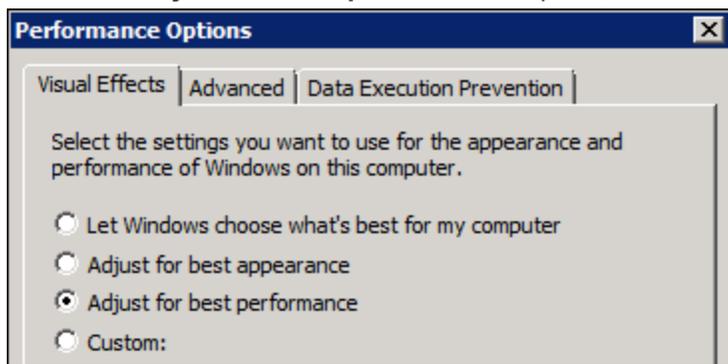
2.4 Windows processor scheduling

Application performance is related to processor scheduling caching options that you set for Windows Server. Processor scheduling determines the responsiveness of applications you run interactively (as opposed to background applications that may be running on the system as services). We recommend that you select background services on all Relativity servers.

1. Select **Settings** in the Performance panel.

2. Select the **Advanced** tab in the Performance Options dialog.

3. In the Processor Scheduling panel, you have the following options:

   - **Programs** - give the active application the best response time and the greatest share of available resources. Use this option only on development servers. You should also select this option when you are using Windows Server 2008 R2 as your desktop operating system.

   - **Background Services** - give background applications a better response time than the active application. Use this option for production servers.

4. Select **Background Services**, and then click **OK**. This does not apply to workers, as workers

should be kept on the Programs setting.
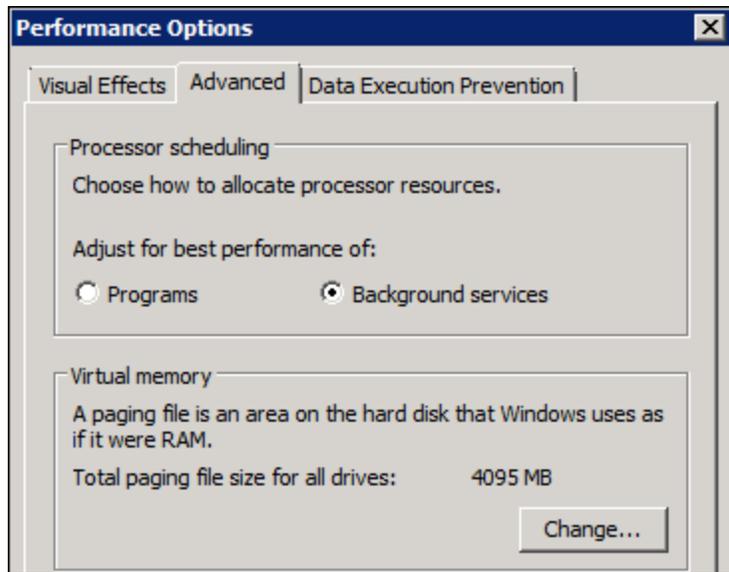
2.5 Windows NTFS fragmentation

Install an automatic disk defragmentation tool on all Relativity Web, SQL, Search, and File servers. There are a number of tools available to defragment your hard drives and optimize performance.

- Some of the latest products are proactive and prevent much of the fragmentation from ever occurring. Research the available tools to decide which is best for you. Unlike the available Windows task to defragment your physical disks, these tools work automatically in the background. These tools use only idle resources to ensure nothing is negatively impacted. Most current SAN and NAS devices include technologies to avoid or limit fragmentation on the block level. Windows sees the data logically from the software level, outside of the storage realm. If Windows detects a file in hundreds of pieces, SAN performance may be affected.

  Note: Work with your storage vendor to see if they recommend installing a defragmenting tool.

- If you don't virtualize the Relativity roles, there will likely be a mirrored array housing the OS and required Relativity components (unless you're also booting from the storage device). If these disks are heavily fragmented, roles including the web servers (IIS) may experience poor performance. This can impact Relativity review.

2.6 Configuring virtual memory

RAM is a limited resource, whereas virtual memory is, for most practical purposes, unlimited. There can be many processes, each one having its own 2 GB of private virtual address space. When the memory that's in use by all existing processes exceeds the amount of available RAM, the operating system moves pages (4 KB pieces) of one or more virtual address spaces to the hard disk. This frees that RAM frame for other uses. Windows stores these "paged out" pages in one or more files. The name of the file is pagefile.sys. You can find this file in the root of a partition. One pagefile.sys file can exist in each disk partition.

You can configure the location and size of the page file in the Control Panel. To set these values, click **System** > **Advanced system settings**, and then click **Settings** under Performance.

By default, Windows Server puts the paging file on the boot partition where the operating system is installed. Windows Server creates a default size of the paging file that is 1.5 times the physical RAM, up to a maximum of 4095 MB.

Consider the following:

- For all Relativity servers, manually set the size of the paging file to 4095 MB. We recommend the size of the paging file since OS volume has only enough room for requirements. OS volume can't support a page file size of 1.5 times the amount of physical RAM.

- For servers with a large amount of RAM installed (16GB+) and more than just the OS volume, create a second page file. Place this page file onto a drive other than the one housing the OS. Set this second page file to a size of 1.5 times the amount of physical RAM. This should be no greater than 50 GB. Microsoft has no specific recommendations about performance gains for page files larger than 50 GB. An example of this server type might include the Analytics server.

- SQL Server shouldn't use the page file for memory on correctly configured servers. There should be no need to create a second page file on the SQL Server. SQL Server memory configurations are detailed in the SQL Server setup section of this guide.

Manually setting the size of the paging file provides better performance than the server automatically sizing it. It's a best practice to set the initial minimum and maximum size settings for the paging file to the same value. This ensures no processing resources are lost to the dynamic resizing of the paging file. This is especially true given that this resizing activity typically occurs when the memory resources on the system are already constrained. Setting the same minimum and maximum page file size values also ensure the paging area on a disk is one single, contiguous area. This improves disk seek time.

Microsoft recommends isolating the paging file onto one or more dedicated physical drives. Configure these drives as either RAID-0 (striping) or RAID-1 (mirroring) arrays. Or the paging file should be on single disks without RAID. Redundancy is not normally required for the paging file. Don't configure the paging file on a RAID 5 array. It's not recommended to configure the paging file on a RAID 5. This is because paging file activity is write-intensive and RAID 5 arrays are better suited for read performance than write performance.

2.7 HTTP2 Configuration

Due to an existing defect with HTTP2, you must add two Windows registry values.

Under `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\HTTP\Parameters`, add these two values of REG_DWORD type:

- **EnableHttp2Tls** with the value of 0
- **EnableHttp2Cleartext** with the value of 0

After you add the values, restart the machine.

2.8 Antivirus directory exclusion list

Configure all antivirus software on any of your Relativity servers to exclude the following areas:

- **SQL Servers** - all directory values are located in the EDDS database Instance Setting table (excluding the Relativity Program Files and the BCPPath). Relativity is installed in the following location: C:\Program Files\kCura Corporation\Relativity. Adjust the drive location if necessary.
  - Directory where the Database files are located.
  - Directory where the Log files are located.
  - The location of the BCPPath on the SQL Server.
  - Directory where the Full Text Indexes are located

- **Agent servers**

  > **Note:** Keep in mind your environment may differ slightly.

  - Default: C:\Program Files\kCura Corporation\Relativity
  - C:\Windows\Temp
  - C:\Users\Rel_SVC\AppData\Local\Temp
  - Account running the agent service = Rel_SVC

- **Web servers**

  > **Note:** Keep in mind your environment may differ slightly.

  - Default: C:\Program Files\kCura Corporation\Relativity
  - C:\Windows\Temp
  - C:\Users\Rel_SVC\AppData\Local\Temp
  - Account running the agent service = Rel_SVC

- **Worker servers**
  - C:\Program Files\kCura Corporation
  - C:\Windows\System32 (\Invariant.hook.dll)
  - C:\Windows\SysWOW64 (\Invariant.hook.dll)
  - C:\Windows\System32\Tasks\Invariant\Relativity Processing Launcher
  - Optional:
    - C:\Users\Rel_SVC\AppData\Local\Temp
    - C:\Windows\Temp

  > **Note:** We recommend that you scan any raw data for malware before introducing it into your Relativity environment. If you perform a scan, you may be comfortable excluding the Temp directories on your worker servers from your antivirus scans as well. This results in better performance and fewer interruptions due to live scans on the temp files that Processing (Invariant) creates in these directories. Unscanned raw data has potential to introduce harmful files into your environment.

- **Analytics indexes** - locate a folder named Content Analyst or CAAT; this is the installation directory. The index directory, if different, should also be excluded from Anti-Virus.

- **Data Grid Data location** - if Elasticsearch is installed, locate this in the Data Grid data node YML file.

- **Elasticsearch service** - \RelativityDataGrid\elasticsearch-main\bin\elasticsearch-service-x64.exe

- **File repositories** - any file directory that Relativity uses as a file share.

- **ARM archive locations** - directory where the ARM archive locations are located.

- **Cache locations** - directory where servers temporarily store converted copies of natives, images, productions, and other file types.

2.9 Management server

Whenever possible, avoid logging in to a production server using remote desktop. Instead, use a management or utility server. This server or virtual machine should have SQL Server Management Studio (SSMS) and the Relativity Desktop Client installed. Use this server to connect to the SQL instances to adjust maintenance plans and query tables. If you have an external hard drive containing data that you want to import into Relativity, connect it to this server and launch the Desktop Client on it to perform the data imports or export.

We also recommend that you install the Marvel cluster to the management server. Use this cluster to monitor and report on the performance and usage of the Data Grid Cluster. The Marvel cluster saves the daily indexes to its own data node. Carefully monitor the drive space used by these indexes so that the drive doesn't run out of space.

When administrators log directly into a production server, navigate and open applications, memory is consumed and processes are wasting CPU cycles. There is also a Windows Server 2008 and R2 issue related to file caches having no cap. System admins should avoid dragging and dropping files via RDP from the console as this action is cached and may result in no free memory being available to the operating system.

3 Configuring SQL Server

Use the following guidelines to configure the SQL Server(s) for optimum performance with Relativity. Some of these configuration options are one-time settings, while others require intermittent updating as your hardware or database sizes change.

Note: These guidelines are applicable to all SQL Servers in the environment, including the Worker Manager SQL Server.

3.1 Optimize for ad hoc workloads option

On the SQL Server, the optimize for ad hoc workloads option improves the efficiency of the plan cache for workloads that contain many single-use ad hoc batches. When this option is set to on (true), the Database Engine stores a small compiled plan stub in the plan cache when a batch is compiled for the first time. (This step is performed instead of the full compiled plan.) It relieves memory pressure by preventing the plan cache to fill with compiled plans that are not reused.

The compiled plan stub allows the Database Engine to recognize that this ad hoc batch has been previously compiled, but has only stored a compiled plan stub. When this batch is invoked (compiled or executed) again, the Database Engine compiles the batch, removes the compiled plan stub from the plan cache, and adds the full compiled plan to the plan cache.

Note: This information was obtained from the following Microsoft article: http://msdn.microsoft.com/en-us/library/cc645587.aspx.

We recommend enabling the optimize for ad hoc workloads option on each SQL instance.

Execute the following statement to enable this option:

```
SP_CONFIGURE 'show advanced options',1
RECONFIGURE
GO
SP_CONFIGURE 'optimize for ad hoc workloads',1
RECONFIGURE
GO
```

3.2 Max degree of parallelism option

SQL Server detects the best degree of parallelism. Parallelism is the number of processor cores employed to run a single statement for each parallel plan execution. You can use the max degree of parallelism option to limit the number of processor cores to use in parallel plan execution. To enable the server to determine the maximum degree of parallelism, set this option to 0 (this is the default value). Setting maximum degree of parallelism to 0 ensures SQL Server uses all available processors (up to 64 processors).

Note: This information was obtained from the following Microsoft article: https://msdn.microsoft.com/en-us/library/ms189094(v=sql.120).aspx.

When it comes to servers with more than one NUMA node, refer to the following: https://support.microsoft.com/en-us/kb/2806535.

We recommend the following MaxDOP settings:

- **Server with single NUMA node** - If the total number of logical processor cores is greater than or equal to 16, set the MaxDOP to a value of 8.

- **Server with single NUMA node** - If the total number of logical processor cores is less than 16, set the MaxDOP to a value equal to half the number of available cores.

- **Server with multiple NUMA nodes** - If the total number of logical processor cores per NUMA node is 8 or more, set MaxDOP to 8.

- **Server with multiple NUMA nodes** - If the total number of logical processor cores per NUMA node is less than 8, set MaxDOP to a value equal to the number logical processor cores per NUMA node.

Another way to determine the optimal setting is to understand the amount of simultaneous long running queries that will be executed at any given time. For instance, in an 16-core system that has 100 simultaneous users logged in executing queries, it might make sense to set MaxDop to a value of 4 instead of 8 to more efficiently support many simultaneous long running queries. Analysis of the SQL Server Dynamic Management Views can assist in making this determination.

This setting takes effect immediately, without restarting the MSSQLSERVER service.

Execute the following statement to set this value.

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE WITH OVERRIDE;
GO
sp_configure 'max degree of parallelism', 8;
GO
RECONFIGURE WITH OVERRIDE;
GO
```

The Optimize for Ad hoc Workloads and Max Degree of Parallelism options can also be viewed and adjusted in the properties of the SQL Server instance within SQL Server Management Studio.

To view and adjust these options:

1. Launch SQL MGMT Studio.

2. Right-click on the Relativity SQL instance.

3. Select **Properties**.

4. Select **Advanced**.
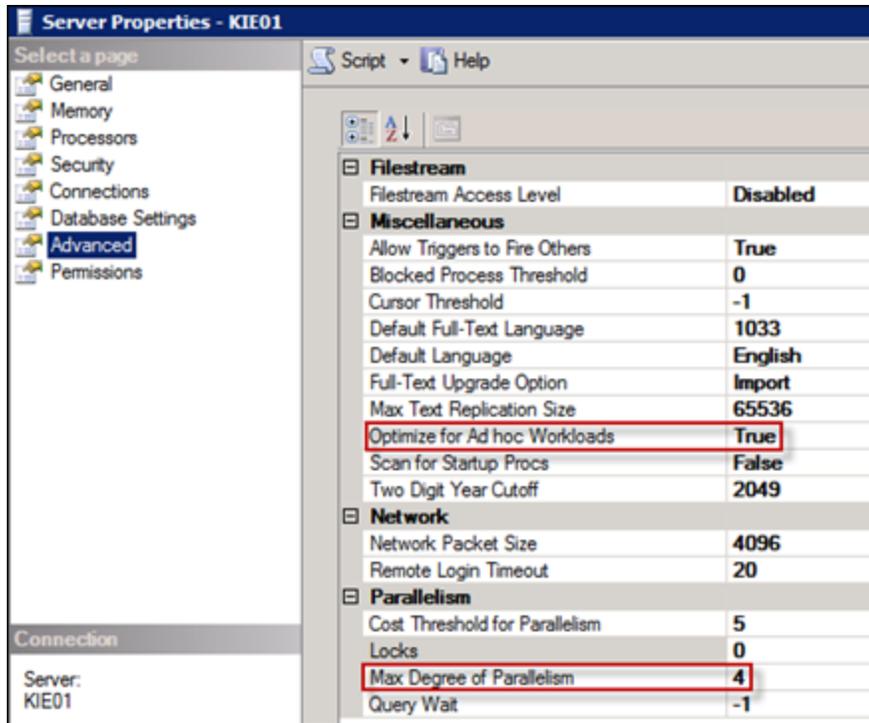
3.3 Cost threshold for parallelism

Use the cost of threshold for parallelism option to specify the threshold at which Microsoft SQL Server creates and runs parallel plans for queries. SQL Server creates and runs a parallel plan for a query only when the estimated cost to run a serial plan for the same query is higher than the value set in cost threshold for parallelism. The cost refers to an estimated elapsed time required to run the serial plan on a specific hardware configuration. The default value is 5.

Note: This information was obtained from the following Microsoft article: http://msdn.microsoft.com/en-us/library/ms188603%28v=sql.105%29.aspx.

We recommend increasing this option to a value of 50 on each SQL instance.

Execute the following statement to set this option:

```
sp_configure 'show advanced options', 1;
GO
reconfigure;
GO
sp_configure 'cost threshold for parallelism', 50;
GO
reconfigure;
GO
```

3.4 Optimize the TempDB database

The tempdb is a database for temporary user and internal objects created by SQL Server. For optimal system performance and stability, ensure the tempdb database is configured correctly.

Use the following guidelines to configure the tempdb database:

- Create one temporary database file per each processor core. Multiple files provide better I/O performance and less contention on the global allocation structures. For higher end systems with more than 8 cores, Microsoft suggests beginning with 8 database files, and adding more if there is still contention. There are scripts available to measure contention; most Relativity environments work well with 8 data files.

- Set the initial size of each file to 10 GB or higher (to limit any autogrowth) and either set autogrowth to 10 percent or disable it. To set file sizes, right-click on the tempdb system database, select Files, and make additions. Set all files to the same initial size. It's important that all data files are set to the same initial size and autogrowth settings. As the environment grows, in an optimized setup, the Tempdb system database should be larger than your biggest, active Relativity workspace eddsdbo.Document table.

- Store these files on a fast RAID array and not the OS partition. RAID 10 is ideal. Some partners have opted to store these data files on PCIe-flash or SSDs. Read the vendor specific instructions for these newer technologies, as they might require free memory for drivers or free disk space for performance.

- Store the log file (.ldf) for this database on the same array as the other database log files. Set the initial size of the log file to 25% of the total of all tempdb data files. For example, if there are 8 tempdb data files set to an initial size of 10 GB each, set the initial size of the tempdb log file to 20 GB. Do not create additional log files – only one is necessary.

Additions can be made while the system is live, but you need to restart the SQL instance for changes to take effect. Make this change on all Relativity SQL Servers.

> **Note:** For more information about the tempdb, see http://msdn.microsoft.com/en-us/library/ms190768.aspx.

Don't install the default tempdb data and log files on the C (OS) partition. Moving the default tempdev and templog files to another partition is important, even though it may not be easy to change their location using SQL Server Management Studio (SSMS). To move these files to a different location, you have to execute some SQL statements in SSMS.

For more information, see http://msdn.microsoft.com/en-us/library/ms345408.aspx.

3.5 SQL maximum server memory

Use maximum server memory to prevent the SQL Server buffer pool from using more than the specified amount of memory. The remaining memory is available to the operating system and other installed applications.

The suggested configuration involves leaving at least 10% of the total amount of RAM available to the OS and other applications. For example, set the SQL Server maximum server memory to a value of 58GB or

lower if there is 64GB RAM available. This assumes that the server is dedicated to SQL and there are no other applications installed that require a significant amount of memory. Some servers benefit from leaving more than 10% available to the OS and other applications.

To set the max server memory option, adjust the max server memory value below, and then execute in SSMS.

```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'max server memory', 1024;
GO
RECONFIGURE;
GO
```

You can also view the maximum server memory option and adjust this option in the properties of the SQL Server instance within SQL Server Management Studio.

To view and adjust the maximum server memory option:

1. Launch SQL MGMT Studio.

2. Right-click on the Relativity SQL instance.

3. Select **Properties**.

4. Select **Memory**.
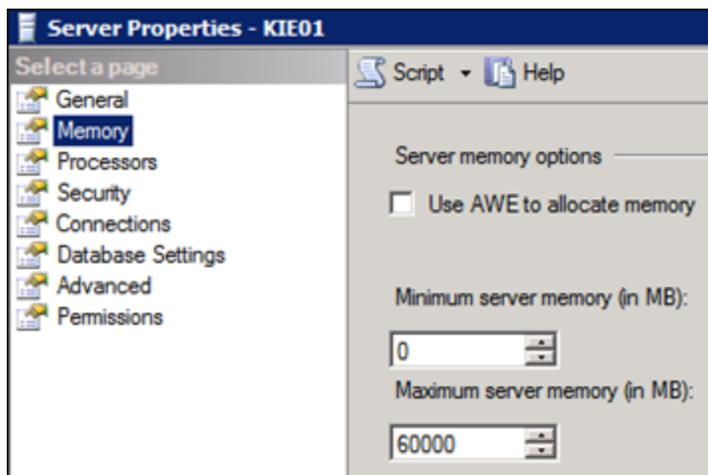
In the event of a failover in an active/active SQL cluster, reduce the maximum server memory value for each instance to half, so they aren't competing for resources (which can severely slow down or crash the server). For example, with 32 GB installed on each node, set the max server memory on each instance to 14 GB (so together they total 28 GB). This leaves 4 GB of memory available for the OS and other installed applications on the active node. This is only recommended if both instances are running on the same box at the same time. Customized startup scripts can dynamically detect whether or not multiple instances are running on the same node and adjust the maximum server memory value for each instance accordingly. Contact Relativity support for more information.

Note: For more information on max server memory, see http://msdn.microsoft.com/en-us/library/ms178067.aspx.

3.6 Instant file initialization

Data and log files are initialized to overwrite any existing data left on the disk from previously deleted files. Data and log files are first initialized by populating the files with zeros when you perform one of the following operations:

- Create a database

- Add log or data file to an existing database

- Increase the size of an existing file (including autogrow operations)

- Restore a database or filegroup

File initialization causes these operations to take longer. However, when data is written to the files for the first time, the operating system doesn't have to populate the files with zeros.

In the SQL Server, data files can be initialized instantaneously. This process creates fast execution of file operations listed above. Instant file initialization reclaims used disk space without populating that space with zeros. Instead, disk content is overwritten as new data is written to the files. Log files cannot be initialized instantaneously.

> **Note:** This information was obtained from the following Microsoft article:
> http://technet.microsoft.com/en-us/library/ms175935.aspx.

To enable instant file initialization:

1. Click **Run** on the Start menu, and then enter **secpol.msc**.

2. Expand **Local Policies**.

3. Select the **User Rights Assignmen**t folder. The policies display in the details pane.

4. Double-click **Perform Volume Maintenance Tasks** in the pane.

5. Click **Add**in the Local Security Policy Setting dialog.

6. Add your SQL Server service account (or a domain group) in the Select Users or Groups dialog.

7. Restart the SQL service or server.

3.6.1 Comparison of zero and instant initialization

The following sample performance tests compare zero initialization and instant initialization (source: http://www.sqlskills.com/BLOGS/KIMBERLY/post/Instant-Initialization-What-Why-and-How.aspx#ixzz0sMB4W7PI).

Performance test	Zero initialization	Instant initialization
Hardware	Dell Precision 670 Dual Proc (x64) with Dual Core, 4 GB Memory, RAID 1+0 array w/4-142 GB, 15000rpm disks.	Dell Precision 670 Dual Proc (x64) with Dual Core, 4 GB Memory, RAID 1+0 array w/4-142 GB, 15000rpm disks.

Performance test	Zero initialization	Instant initialization
Create Database with 20 GB Data file	14:02 minutes	1.3 seconds
Alter database by 10 GB	7:01 minutes	4 seconds
Restore 30 GB database (empty backup)	21:07 minutes	5 seconds
Restore 30 GB database (11GB backup)	38:28 minutes	19:42 minutes

3.7 File allocation unit size

File Allocation unit size is also called the Cluster size. Windows NTFS formatted drives support sizes of 512, 1024, 2048, 4096, 8192, 16K, 32K and 64K. The allocation unit is the smallest amount of space that a file can consume. Windows uses a default allocation unit size of 4096 bytes if this isn't set during the format. Unfortunately, this is usually not ideal for SQL Server performance. In most scenarios, the Data and Tempdb drives should be formatted with a 64K Allocation Unit Size. Configure this early in the SQL Server setup process as the allocation unit size is set during a drive format operation.

Run the following command for each drive to see the file allocation unit size reported in bytes per cluster. The Bytes Per Cluster value, which contains the file allocation unit size, is highlighted here.

For example, to run the command for E: drive:

```
fsutil fsinfo ntfsinfo e:
```

To run the command for F: drive:

```
fsutil fsinfo ntfsinfo f:
```

An appropriate value for most installations is 65,536 bytes (that is, 64 KB) for partitions on which SQL Server data files reside. If necessary, consult your storage vendor for their recommendation or assistance with this configuration. This significantly impacts SQL Server performance.

Always use the storage vendor's recommended setting. If your storage vendor doesn't have a recommended setting, use 64 KB.

Note: This information was obtained from the following article: http://msdn.microsoft.com/en-us/library/dd758814.aspx.

3.8 Microsoft SQL Server service packs

Ensure you have the latest Microsoft SQL Server Service Pack installed on all Relativity SQL Servers. Microsoft uses service packs to distribute hotfixes and software updates for the SQL Server. In addition, these service packs may include system admin tools, drivers, and additional components. These updates and components are conveniently bundled for easy downloading.

Since service packs are cumulative, each new one contains any recent fixes as well as any previous updates already released. You don't have to install a previous service pack before you install the latest one.

3.8.1 Determining the version of SQL Server

To determine which version of Microsoft SQL Server is running in your environment:

1. Connect to the SQL Server with SQL Server Management Studio.

2. Run either of the following Transact-SQL statements:

```
SELECT SERVERPROPERTY('productversion'), SERVERPROPERTY ('productlevel'), SERVERPROPERTY
('edition')
SELECT @@VERSION
```

3.9 Autogrowth settings

SQL Server automatically increases the size of database (.mdf) and transaction log (.ldf) files throughout the life of a workspace. It is important to set these auto growth variables appropriately.

If a transaction log is set to auto grow by the default 10% on a very large file, it takes time for the grow operation to complete. Other transactions that require a write to the transaction log have to wait until the grow operation finishes. Conversely, if these files are set to expand by too small increments, higher disk fragmentation occurs in addition to the server being continuously taxed by having to expand these files.

Relativity data files are by default set to autogrow by 10 percent and their logs by 512 MB as specified in the Autogrowth setting. Don't set the transaction logs (.ldf) to auto grow by more than 1024 MB, because log files can't be initialized instantaneously.

Note: Databases enabled for Transparent Data Encryption (TDE) cannot take advantage of Instant File Initialization. Therefore, you should set databases enabled for TDE to autogrow by 4096MB instead of the default 10%. This is because if a database is large and needs to autogrow by 10% without being able to take advantage of Instant File Initialization, it can result in application timeouts during each autogrow.

If possible, try to anticipate how large a database will become and set the "Initial Size" of the MDF file within the properties of that database to avoid or limit automatic growth.

Note: When a new workspace is created in Relativity, the template workspace database is backed up and restored as the new workspace database. The autogrowth settings from the backed up database will carry over to the new workspace.

3.10 SQL Server best practices analyzer

The Microsoft SQL Server BPA is a diagnostic tool that performs the following functions:

- Gathers information about a Server and a Microsoft SQL Server instance installed on that server.

- Determines if the configurations are set according to the recommended best practices.

- Reports on all configurations, indicating settings that differ from recommendations.

- Indicates potential problems in the installed instance of SQL Server.

- Recommends solutions to potential problems.

We recommend installing this on each Relativity SQL Server. Perform a diagnostic check and review the report prior to introducing a SQL Server to the environment. Perform another analysis periodically.

3.11 RAID levels and SQL Server

There are many iSCSI and Fibre channel SAN, DAS and NAS solutions available. These devices often utilize RAID levels not discussed in this guide and include other unique technologies to provide the fastest I/O possible. Work with your storage unit provider to ensure the device is configured for optimal performance. Work with your storage unit provider or follow Microsoft best practices to determine where different SQL data should live.

- The tempdb data files should reside on the fastest disks available, separate from that of all other user databases. You must configure the tempdb database correctly for SQL Server performance. Many clients have opted to store these data files on PCIe-flash or SSDs as the data is temporary and doesn't require shared storage in SQL Server Failover Cluster setups. Regardless of the type of SAN or DAS and any caching mechanisms or auto-tiering, storing these on local PCIe-flash or SSD RAID will result in reduced latency and take some of the load off of the primary storage device (s).

- Database log files should usually reside on their own disks. Isolating the database log files on their own spindles helps to ensure that nothing interferes with the sequential writes for each database log file.

- Solid state storage is an exception to the conventional wisdom. Hard drives perform very differently depending on whether you are doing random or sequential operations. Random operations are slower because of the moving parts included in typical hard drives. This is why you may want to avoid placing the log and data files on the same disks. Solid state disks, however, typically perform the same whether random or sequential operations are performed.

As mentioned above, many storage solutions include unique technologies to provide the fastest I/O possible that may go against the conventional wisdom. Many newer SAN devices are intelligent enough to automatically shift data across different types of available storage depending on activity levels of the stored data. These devices can help you maximize performance with minimal manual intervention, reducing IT related costs and getting as much possible out of the available storage.

Regardless, there are many different storage options available and we encourage you to research and decide which solution is best for your organization.

Below are some high-level suggestions for our smaller 25-50 user licensed instances that may not include a SAN and are looking for some guidance in configuring local SQL storage. RAID (Redundant Array of Independent Disks) levels 0, 1, 5 and 1+0 are typically implemented with SQL Server. Each RAID level below outlines the SQL data pieces they typically support.

3.11.1 Level 0

Use for: PageFile, TempDB

This level is also known as disk striping because it uses a disk file system called a stripe set. Data is divided into blocks and spread in a fixed order among all disks in an array. RAID 0 improves read and write performance by spreading operations across multiple disks. Operations can then be performed independently and at the same time.

We would never suggest RAID 0 for anything as it can introduce system or application specific outages. There is no fault tolerance with RAID 0. It improves performance, but doesn't provide redundancy.

3.11.2 Level 1

Use for: OS, Logs (.ldf)

This level is also known as disk mirroring because it uses a disk file system called a mirror set. Disk mirroring provides a redundant, identical copy of a selected disk. All data written to the primary disk is written to the mirror disk. RAID 1 provides fault tolerance and generally improves read performance but may degrade write performance.

3.11.3 Level 5

Use for: Databases (.mdf), Full Text Indexes (.ndf)

Also known as striping with parity, this level is the most popular strategy for new designs. RAID 5 is similar to RAID 4, because this level stripes the data in large blocks across the disks in an array. However, RAID 5 differs because it writes the parity across all the disks. Data redundancy is provided by the parity information.

The data and parity information are arranged on the disk array so that the two types of information are always on different disks. Striping with parity offers better performance than disk mirroring (RAID 1). However, when a stripe member is missing, read performance is decreased, for example, when a disk fails.

3.11.4 Level 10 (1+0)

Use for: Databases (.mdf), Full Text Indexes (.ndf), TempDB, Logs (.ldf)

This level is also known as mirroring with striping. RAID 10 uses a striped array of disks that are then mirrored to another identical set of striped disks. For example, a striped array can be created by using five disks. The striped array of disks is then mirrored using another set of five striped disks. RAID 10 provides the performance benefits of disk striping with the disk redundancy of mirroring. RAID 10 provides the highest read-and-write performance of any one of the other RAID levels but at the expense of using two times as many disks.

RAID levels higher than 10 (1 + 0) may offer additional fault tolerance or performance enhancements. These levels generally are proprietary systems. For more information about these types of RAID systems, contact the hardware vendor.

The following table breaks down the various RAID levels (source: http://technet.microsoft.com/en-us/library/cc966414.aspx#_RAID).

RAID Levels	RAID 0	RAID 1	RAID 5	RAID 10
Reliability	Lowest - Lack of fault tolerance results in data loss.	Very good - Even better with duplexing.	Good - Can tolerate single machine fault.	Excellent

RAID Levels	RAID 0	RAID 1	RAID 5	RAID 10
Storage Efficiency	100 percent	50 percent	>50 percent, <100 percent (#drives-1/#drives)	50 percent
Random Read	Excellent	Fair - Worst of the RAID levels but better than a single drive.	Excellent	Excellent
Random Write	Excellent	Fair - Worse than a single drive but better than some RAID levels.	Fair - Generally better with larger stripe sizes.	Very good
Sequential Read	Excellent	Fair - Comparable to a single drive.	Very good. Generally, better with smaller stripe sizes.	Very good
Sequential Write	Excellent	Good - Better than other RAID levels.	Fair	Very good
Cost	Lowest	Moderate - Relatively high cost due to redundant drives; however, no expensive controller required.	Moderate	High
Recommended use	Good for non-critical data or stag-	Good for data that requires	Very good for Read only data.	Data requiring high performance for

RAID Levels	RAID 0	RAID 1	RAID 5	RAID 10
	nantly updated data that is backed up regularly or any data requiring fast write performance at very low cost. Great for testing.	high fault tolerance at relatively low hardware cost (redundancy using parity requires more expensive hardware). Best for log files.		both read and write and excellent reliability while trading off storage efficiency and cost.

The following table provides a summary of an example twelve disk RAID configuration.

Disk	Qty	RAID	Example RAID-12 disk setup
0	2	1	Windows
1	2	0	tempdb
2	2	5 or 10	Transaction Logs (ldf)
3	4	5 or 10	Relativity Databases (mdf)
4	2	5 or 10	Full Text Catalogs (ndf)

4 Configuring the Analytics server

Use the following guidelines to configure your Analytics server(s) for optimum performance with Relativity.

4.1 Memory requirements

4.1.1 Analytics indexing

Server memory is the most important component in building an Analytics index. The more memory your server has, the larger the data sets that can be indexed without significant memory paging. Insufficient memory will slow down index build performance.

The following factors affect RAM consumption during indexing:

- Number of documents in the training set
- Number of documents in the searchable set
- Number of unique words across all the documents in the data set being indexed
- Total mean document size (as measured in unique words)

Use the following equation to estimate how much free RAM is needed to complete an index build:

(Number of Training Documents) * **6000** = **Amount of RAM needed in bytes**

An easy way to remember this equation is that every 1 million training documents in the index require 6 GB of free RAM. The equation is based upon the average document set in Relativity. If your data set has more unique terms than an average data set, more RAM will be required to build. We recommend accounting for slightly more RAM than the equation estimates.

4.1.2 Structured analytics

To run structured analytics, the Analytics server can require substantial server resources. The structured analytics features are run by the Java process, as well as PostgreSQL. One of the most important parts of making sure a structured analytics operation succeeds is ensuring that Java has access to enough RAM. This is referred to as Java heap size, or JVM.

Use the following equation to estimate how much JVM will be required for a given structured analytics set:

(Number of Documents) * 6000 = Amount of JVM needed in bytes

An easy way to remember this equation is that every 1 million documents in the set require about 6 GB of RAM for the Java process. If your data set is comprised of very long documents, it may require more JVM. If it is comprised of very small documents, then you may not need as much JVM. If Java does not have enough memory to complete a structured analytics operation, you will sometimes receive an out of memory (OOM) error. More often, though, Java will heap dump and garbage collect endlessly without ever successfully completing the operation. This equation is a good starting point so that these types of problems do not occur. For information on how to configure JVM, see Java heap size (JVM) on the next page.

4.1.3 Enabled Analytics indexes

An Analytics index is stored on disk and is loaded into memory when the index has queries enabled (the **Enable Queries** button on the index). An index with queries enabled may be used for all Analytics functions such as clustering, categorization, and querying. When you enable queries on an Analytics index, Relativity loads the vectors associated with all searchable documents and words in the conceptual space into RAM in an LSIApp.exe process. For indexes with millions of documents and words, this RAM requirement may be thousands of megabytes.

The following ranges indicate the amount of RAM needed to enable queries on an index:

- Number of bytes required (Low end): (Number of searchable documents) * 400

- Number of bytes required (High end): (Number of searchable documents) * 5000

Click **Disable Queries** on any Analytics indexes that aren't in use to free up RAM on the Analytics server. The **MaxAnalyticsIndexIdleDays** instance setting helps with this issue. This value is the number of days that an Analytics index can remain inactive before the Case Manager agent disables queries on it. Inactivity on the Analytics index is defined as not having any categorization, clustering, or any type of searches using the index. This feature ensures that indexes that are not being used are not using up RAM on the Analytics server. If the index needs to be used again, navigate to the index in Relativity and click **Enable Queries**. It will be available for searching again within seconds.

4.2 Java heap size (JVM)

Depending on the amount of RAM on your Analytics server, as well as its role, you will need to modify the Java Heap Size setting. This setting controls how much RAM the Java process may consume. Java is used for index populations, structured analytics operations, clustering, and categorization.

Here are some general guidelines:

- If the Analytics server is used for both indexing and structured analytics, set this value to about 50% of the server's total RAM. You need to leave RAM available for the LSIApp.exe process, which is used for building conceptual indexes.

- If the Analytics server is used solely for structured analytics, set this value to about 85% of the server's total RAM, but leave at least 10 GB of RAM available for the underlying database processes.

- If the Analytics server is used solely for indexing, set this value to about 50% of the server's total RAM. You need to leave RAM available for the LSIApp.exe process, which is used for building conceptual indexes.

- For all configurations, set the Java minimum setting (-Xms) to 12.5% of the total RAM on the server.

Due to a limitation in the Java application, do not configure JVM with a value between 32 GB and 47 GB (inclusive). When JVM is set between 32 GB and 47 GB, the application only has access to 20-22 GB heap space. For example, if the server has 64 GB RAM, set JVM to either 31 GB or 48 GB. This allows the Java application to access all of the allocated RAM.

To modify the Java Heap Size setting, perform the following steps:

1. Navigate to **<CAAT install drive>\CAAT\bin**.

2. Edit the **env.cmd** file.

    1. Locate the line similar to the following: **set HEAP_OPTS=-Xms4096m -Xmx16383m**. The configuration starting with –Xmx refers to the maximum amount of RAM available to Java, in megabytes. In the example, a maximum of 16383 megabytes (16.383 GB) is available.
    The configuration starting with –Xms refers to the minimum amount. In the example, a minimum of 4096 megabytes (4.096 GB) is available.

    2. Modify these values as needed. Both megabyte and gigabyte values are supported for these settings.

3. Stop and start the Analytics Engine (Content Analyst CAAT) service. This allows the changes to take effect.

Note: Never set the Java maximum (-Xmx) to be less than the Java minimum (-Xms).

4.3 Page file size

We recommend the following settings regarding the page file size for the Analytics server:

- Set the size of the paging file to 4095 MB or higher. This is because the OS array generally only has enough room for what's required and is not able to support a page file size of 1.5 times the amount of physical RAM.

- Set the initial minimum and maximum size of settings for the page to the same value to ensure no processing resources are lost to the dynamic resizing of the paging file.

- Ensure that the paging area on a disk is one single, contiguous area, which improves disk seek time.

- For servers with a large amount of RAM installed, set the page file to a size no greater than 50 GB. Microsoft has no specific recommendations about performance gains for page files larger than 50 GB.

4.4 Index directory requirements

The index directory stores both the Analytics indexes and the structured analytics sets. Using default settings, the average amount of disk space for the Analytics index or structured analytics set is equal to about 20% of the size of the MDF file of the workspace database. This metric indicates an average amount of disk space usage, but actual indexes may vary considerably in size. The amount of space required depends on the size of the extracted text being indexed, as well as the number of documents, unique words, and settings used to build the index. An Analytics server may not have multiple index locations; it may only reference one disk location for the server's Analytics indexes and structured analytics sets.

Due to the size requirements, we recommend you don't store indexes on the local drive where the CAAT directory is installed. Upon installation or upgrade, the Analytics Server installer will prompt for the index directory location. If you would like to move the index directory location after upgrade, see Moving Analytics indexes and structured analytics sets in the Admin Guide.

CAAT® uses the database software PostgreSQL which requires guaranteed synchronous writes to the index directory. The connection from the analytics server to the index directory should be one that guarantees synchronous writes, such as Fibre Channel or iSCSI, rather than NFS or SMB/CIFS. We recommend storing the indexes and structured analytics sets on locally mounted storage rather than a remotely mounted file system.

4.5 CPU requirements

4.5.1 Analytics indexing

The Analytics index creation process also depends on CPU and I/O resources at various stages of the build. Ensuring that your server has multiple processors and fast I/O also increases efficiency during the population build process. Adding more CPU cores to an Analytics server ensures that index populations are as fast as possible—especially for large indexes. When you add additional CPU cores, you may also increase the number of Maximum Connectors used per index on the server. To modify this value, see Editing an Analytics server in the Admin Guide.

4.5.2 Structured analytics

The structured analytics features are run by the Java process as well as PostgreSQL. In order for these processes to operate most efficiently, allocate sufficient CPU cores to the Analytics server. For optimal Textual Near Duplicate Identification performance, the Analytics server needs at least 8 CPU cores. Textual Near Duplicate Identification performance improves as additional cores are added.

The following charts illustrate the "Running Analytics Operations" step of a Textual Near Duplicate Identification structured analytics set with the default Minimum Similarity Percentage of 90 in CAAT® 3.17:

Data set 1: Enron data – 1M documents, 6.2 GB total text
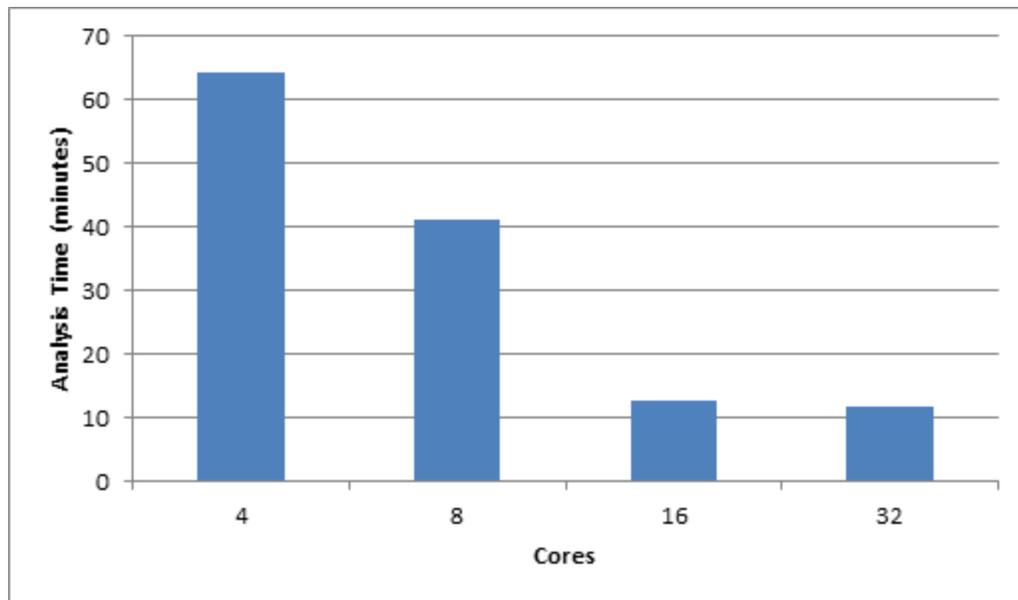
Data set 2: Wikipedia data - 1M records, 2.4 GB total text
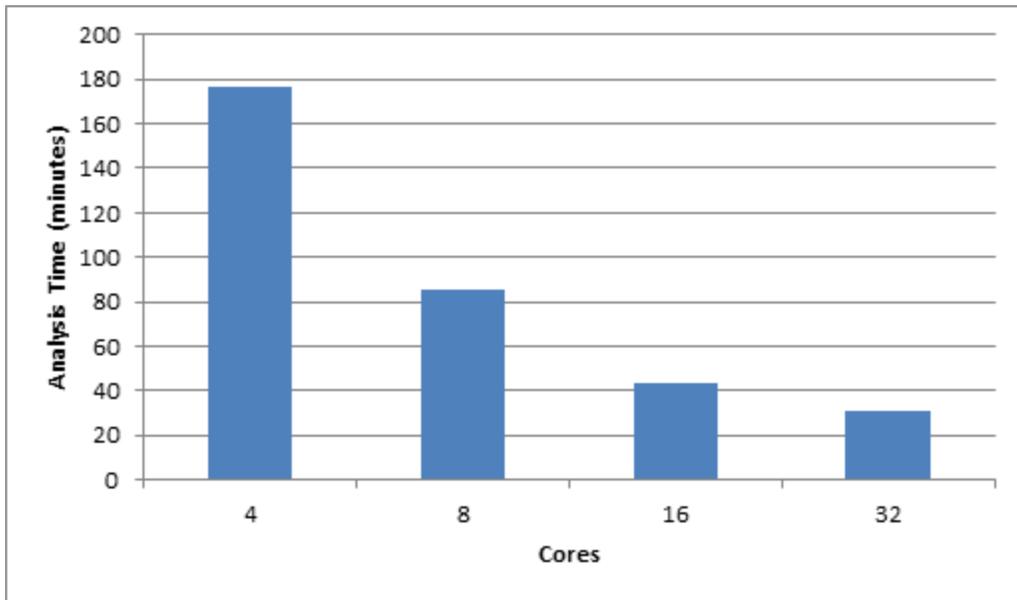
Data set 3: Emails - 768K documents, 11.7 GB total text
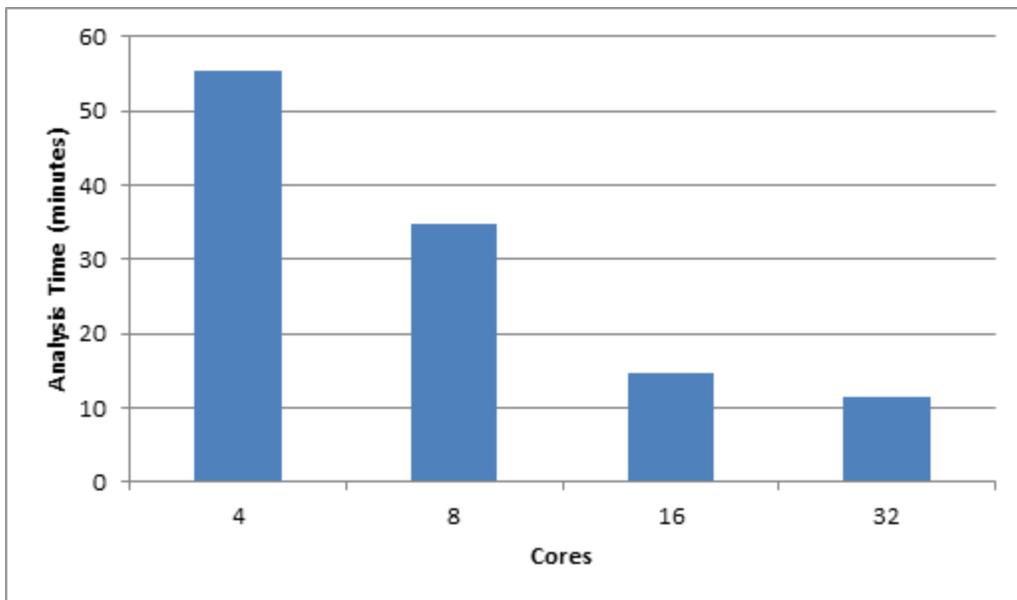
The performance of a given data set varies based on certain factors outside of the number of documents or the total text size. The following types of data sets require more time to analyze:

- Data sets with a very high number of similarly-sized documents

- Data sets with a very low number of textually similar documents

Additionally, if you lower the Minimum Similarity Percentage from the default of 90, more time is required to analyze the data set.

4.6 Scaling

It is often beneficial to add multiple Analytics servers to the Relativity environment. Jobs can run concurrently without adversely affecting each other. It also allows servers to be dedicated to a feature set (structured or indexing) which makes RAM management easier. The following table shows some example environment configurations as well as the typical upper limitation that will be encountered. The upper limitation assumes no other concurrent activity on the server. The upper limit is intended to serve as an estimate and is not a guarantee. Data sets vary widely, and some may require more server resources than usual.

4.6.1 Tier 1 example

For an entry level environment with 100 to 300 users, usually one Analytics server is enough.

Here is an example environment configuration:

Server Name	Role	RAM	JVM	CPU Cores	Upper limit - Structured	Upper limit - Indexing
ANA-01	Both Structured and Indexing	32 GB	16 GB	8	3 million documents	3 million documents

4.6.2 Tier 2 example

For a mid-level environment with over 300 users, you may need to add an additional server to allow for concurrent jobs. Splitting the server roles also allows the servers to work on more data due to the allocation of Java heap.

Here is an example environment configuration:

Server Name	Role	RAM	JVM	CPU Cores	Upper limit - Structured	Upper limit - Conceptual Indexing
ANA-01	Structured	32 GB	22 GB	8	4 million documents	N/A
ANA-02	Indexing	32 GB	16 GB	4	N/A	4 million documents

4.6.3 Tier 3 examples

For a large scale environment, you will likely need to scale up the server and add additional servers. Splitting the server roles allows the servers to work on more data due to the allocation of Java heap. Add more Analytics servers to run more jobs concurrently. However, adding a large amount of RAM to one server will allow a very large job to complete successfully. The balance needs to be determined based on the client needs.

The following are example Tier 3 environment configurations:

Example 1

Server Name	Role	RAM	JVM	CPU Cores	Upper limit - Structured	Upper limit - Conceptual Indexing
ANA-01	Structured	64 GB	54 GB	16	8 million documents	N/A
ANA-02	Indexing	64 GB	31 GB	12	N/A	7 million documents

Example 2

Server Name	Role	RAM	JVM	CPU Cores	Upper limit - Structured	Upper limit - Conceptual Indexing
ANA-01	Structured	128 GB	108 GB	32	16 million documents	N/A
ANA-02	Indexing	128 GB	64 GB	24	N/A	14 million documents

5 Setting up your workspace

As a system admin, use the following best practices to optimize your Relativity environment.

5.1 Fixed-length vs. long text field types

Use the appropriate field type and length for text fields within Relativity.

Fixed-length text - a text field with a limited character length.

- Maximum technical limitation of 4,999 characters

- Examples: Author, Email From, Email Subject

Long Text - a text field larger than 4,999 characters.

- The technical limitation on a long text field is 2 GB.

- Relativity is typically configured to stream files smaller than 50 MB. You can set the maximum file size in the Instance Setting table.

- Only the first 20,000 characters appear in Layouts and Views.

You can create indexes on particular columns to improve performance for queries against large databases. See the Admin Guide for more information.

Consider these SQL limitations when deciding between the two field types: you can't create an index on a column with a type of Long Text, a Unicode-enabled fixed-length text field of more than 450 characters, or a non-Unicode-enabled fixed-length text field of more than 900 characters.

5.2 Fixed-length text field considerations

If there are more than 8,060 characters in a single database row, SQL may experience performance issues. As a result, a best practice is to limit the length of fixed-length text fields.

You can get the total field length usage for any workspace by viewing the document object properties in the **Object Type** tab for that workspace. This also applies to any dynamic objects.
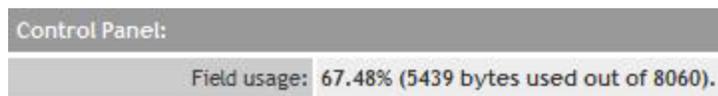
When you create a fixed-length text field, configure the length of the field to the appropriate size. You can increase the length as necessary throughout the life of the workspace. If you have a field that will never exceed a certain length, set the length of the field to that value.

For example, a Document Extension field should have a small field length value. If there is a field that could eventually grow to a great length—a Unicode-enabled fixed-length text field of more than 450 characters or a non-Unicode-enabled fixed-length text field of more than 900 characters—then set it as a Long Text field initially rather than a Fixed Length Text field.

Note: For more information on limiting the total length of text fields in SQL, see http://msdn.microsoft.com/en-us/library/ms186981.aspx.

After any major cleanup or length reduction of fixed-length text fields in a workspace, run the DBCC CleanTable SQL Script after hours on the database to reclaim free space. For the steps on running this script, see http://msdn.microsoft.com/en-us/library/ms174418.aspx.

5.3 Unicode support

Define the Unicode Enabled property of a field prior to importing data. A database level operation to convert a field's data from Unicode to non-Unicode (or vice versa) can take a long time, lock tables, and potentially timeout if performed on a large data set in Relativity.

Only the following field types are Unicode-compatible:

- Fixed-Length Text
- Long Text
- Single Choice
- Multiple Choice

Define the Extracted Text field as Unicode if there is a possibility that text in languages other than English will be loaded into the workspace.

5.4 Data imports

Use the Relativity Desktop Client or Processing to import data into Relativity.

There are two methods of importing native/TIFF files when using the Relativity Desktop Client:

- Copying files from the original location in the selected load file field to the document repository. This copies the files from the original location in the selected load file field to the selected document repository.
- Importing files that already reside in a valid, Relativity-accessible location.

  Note: Use this option when native files already have been copied to their final location, accessible by Relativity. This location should be separate from that of the Relativity document repository. You can set the default option on an environment-wide level. For more information, see the Relativity Instance Setting Guide.

For either method, the selected native file path field should point to the current location of the native files.

Many clients use the second option to reduce the total time it takes to load files into the Relativity repositories. You can manually copy files to where they should reside and then provide the Relativity Desktop Client with pointers to their locations. This can save significant time when loading large amounts of natives and/or TIFFs.

Note: Never run the Relativity Desktop Client on a Relativity production server (Web, Agents, SQL, or Search). Launch the tool on a different server or workstation to prevent resource contention.

5.5 Analytics

Use the following guidelines when setting up an Analytics index in your workspace.

- When populating an Analytics index, exclude non-conceptual files such as XLS, EXE, or RAR from the index. Make sure that the Training Set is limited to documents with less than 2 MB of Extracted Text.

- Disable queries on indexes that are no longer being actively used. Queries are automatically disabled for indexes that haven't been accessed within a pre-set number of days.

  Note: This value is configured using the MaxAnalyticsIndexIdleDays value in the EDDS.EDDSDBO.InstanceSetting table.

- When queries are enabled on an Analytics index, the index is loaded into RAM. Unused indexes shouldn't consume resources on the CAAT server.

- Use the following guidelines when setting up a structured analytics set in your workspace.

5.6 Views and searching

Follow these best practices for views and searching:

- Avoid **Is Like** and **Is Not Like** statements on un-indexed queries. Full table scans on large databases using Is Like statements severely load the server. A full table scan occurs when a query reads every row in a table in order to locate the required information.
  If possible, use a **Contains** condition instead. In order to use Contains, you must add the field to the Full Text Index.

  Note: Is Set and **Is Not Set** criteria also triggers a full table scan. A full table scan operation can be very IO-intensive and take a while to complete on slower storage units.

- Prevent users from filtering and/or sorting on Long Text fields within the document list in large workspaces. To do this, make sure the field's Allow Sort/Tally property is set to **No**.

5.7 Tally/Sum/Average mass operation

In the Tally/Sum/Average mass operation, the Tally option is audited. For the set of requested documents, it lists each unique value found in that field along with the frequency of each occurrence. The process can run on Fixed Length Text, Choice, User, and Number fields.

Tallying on un-indexed columns in large workspaces can take a while to complete and slow down database/system performance. Be sure to place indexes on columns in SQL that are regularly tallied in large workspaces.

You can disable the ability to Tally in a workspace by setting each field's Allow Sort/Tally property to **No**.

5.8 Group by for Pivot

When performing Pivot operations in large workspaces, you can improve Pivot performance by placing a non-clustered index on the field being grouped on. Use SQL Server Management Studio to place the index.

5.9 User notifications

There are several notifications available to help system admins better manage their Relativity environments.

5.9.1 Disclaimer Message

When you want users to agree to a disclaimer message when using Relativity, you can create a custom message and an agreement button for users when they first log in to the Relativity environment. This type of message is useful to ensure that users acknowledge Relativity terms of use or the confidential nature of Relativity content.

To customize a disclaimer on a Relativity login page, use the following steps:

1. Navigate to the web server(s) at C:\Program Files\Relativity Corporation\Relativity\EDDS.

2. Locate the text file, Disclaimer.txt, and update it with your disclaimer text. You must use plain text only; HTML will not display here.

3. Update the DisplayDisclaimer instance setting value to true to display the disclaimer message.

5.9.2 Message of the Day (MotD)

The Message of the Day (MotD) is a message displayed to all users when they log in to Relativity. MotD is most commonly used to inform users of planned system maintenance.

To activate or change the message of the day, navigate to the Instance Details tab from Home.

5.9.3 User status

The user status page displays a list of users currently logged into the system. To access the user status page, navigate to the **User Status** tab from Home.

You can also send messages to any logged in users. This is helpful if you require emergency downtime during a review. System admins can also force log out users from the system on this page.

5.9.4 Default workspace tabs

When a user enters a workspace within Relativity, they are usually taken to the **Documents** tab. You can change this default setting in the **Tabs** tab of any workspace.

Set the **Is Default** property to Yes for the tab that you want to designate the default tab in a workspace. If a user doesn't have access to the Default tab, he or she will be directed to the Documents tab.

Modifying the default tab may be helpful in informing users of any upcoming workspace-level maintenance. Additionally, system admins can direct users to custom pages that contain links to instructional items, important information about the matter, or an overview of workspace review progress.

6 Monitoring environment performance

The hardware and software hosting the instance must be healthy to provide an exceptional Relativity experience. To ensure that your environment is healthy, review the following performance considerations.

6.1 Windows and SQL Server log analysis

Regularly monitor Windows and SQL event logs. These tools can automatically email your IT team when warnings or errors occur. This type of monitoring is often overlooked by IT administrators. It's important to actively monitor these logs on a regular basis in the environment.

There are many tools available to monitor and notify system admins of any issues that arise. It's crucial that all Relativity servers are monitored for the following:

- **Available disk space** - Alert system admins of low disk space on any Relativity server.

- **Server availability** - Alert system admins if any server becomes unresponsive. This is accomplished with simple internal ping tests.

- **Website availability** - Alert system admins if the website becomes unavailable.

- **Resource utilization** - See Resource utilization below.

6.2 Resource utilization

Monitoring resource utilization is a more involved process. Using this process, you can identify bottlenecks and anticipate when it might be time to scale or adjust the environment. The license agreement doesn't limit the amount of hardware or Relativity roles you can add to an instance; you can make adjustments as needed.

Monitor processor and memory utilization across all servers. Note that SQL always shows high memory utilization because it stores as much as it can in memory. It's important to identify any SQL Server storage bottlenecks, which includes the disks or the connection to the disks.

Best practices for monitoring resource utilization include gathering benchmarks at different times of the day. This determines what the acceptable ranges are for the performance counters you intend to monitor. Create this baseline so you can compare it to results as the environment continues to grow. Third party monitoring tools often simplify the gathering and summarizing of performance related metrics. Summarizing this material assists your team to more easily identifying when it might be time to scale one or more areas of the system.

It's difficult to provide specific thresholds or recommendations for performance counters as every environment is different.

6.2.1 Monitoring disk usage

Disk latency is the fundamental measure of disk performance.

- The Avg. Disk/sec Read and Avg. Disk/sec Write counters of the Windows Performance Monitor "Logical or Physical Disk" Objects. You can use these to measure disk latency.

- SQL Server Avg. Disk/sec Read latency should generally be 20ms or less per Microsoft best practices for database volumes. However, 100ms or less is more realistic in environments with traditional storage media. Our clients can take advantage of more recent developments in flash and solid storage.

- SQL Server Avg. Disk/sec Write latency should generally be 3-5ms or less per Microsoft best practices for transaction log volumes. However, 20ms or less is more realistic in environments with traditional storage media. Keep the I/O response times on SQL database log files as low as possible.

Monitor the SQL Server tempdb system database in Relativity environments. This often becomes a bottleneck for many of our larger clients. It's more common for clients to store the tempdb database for each SQL instance on flash or solid state storage instead of traditional, spinning media.

The following table provides additional suggestions for optimal SQL Server disk latency.

Object	Counter	You Want	Description
Physical disk	Avg. Disk Sec/Read	< 8 ms	A key measure of disk latency representing average time, in milliseconds, of each read to disk where >20 is poor, <20 is good/fair, <12 is better, <8 is best
Physical disk	Avg. Disk Sec/Write	< 8 ms (non-cached) < 1 ms (cached)	A key measure of disk latency representing the average time, in milliseconds, of each write to disk where non-cached writes (>20 poor, <20 fair, <12 better, <8 best) differ significantly from cached writes (>4 poor, <4 fair, <2 better, <1 best). For OLTP databases, the lower this number the better, especially for disks holding the transaction log.

For additional information on monitoring for Disk I/O bottlenecks in a system, see http://msdn.microsoft.com/en-us/library/ms175903.aspx.

The disk queue counters aren't usually helpful since they require that you know how many spindles are included in an array as well as any available caching or auto-tier mechanisms.

It's also very important to ensure that the link between SQL and the storage is configured for optimal performance. Work with your storage unit provider to ensure that the pathing is set up correctly and provides optimal performance. This setup becomes more complicated when virtualizing SQL. Other performance objects typically monitored for SQL include several memory counters.

Page life expectancy can indicate if SQL has enough memory available to perform optimally. Lower values during regular operations suggest a need to install more memory. A popular suggested threshold for this counter is 300 seconds, which is often far too low. A more realistic value can be determined by performing this equation (DataCacheSizeInGB/4GB *300).

Note: For more information, see http://www.sqlskills.com/blogs/jonathan/finding-what-queries-in-the-plan-cache-use-a-specific-index/.

In addition to other available tools, you can log these performance counters using the Windows Performance Monitor in Windows Server.

6.3 Analytics performance considerations

Use the following guidelines to ensure that the analytics operations perform quickly and efficiently in your Relativity instance.

6.3.1 Server requirements for index builds

Server memory is the most important factor in building an analytics index. The more memory the server has, the larger the datasets can be indexed without significant memory paging. Use the following equation to estimate how much free RAM on the analytics server is needed to complete an Analytics index build:

(Number of training documents) * 6000 = RAM required (bytes)

Note: This equation is based upon the average document set in Relativity. If the dataset has more unique terms than an average dataset, more RAM is required to build the index.

The analytics index creation process also depends on CPU and I/O resources at various stages. Ensuring that the analytics server has multiple processors and fast I/O increases efficiency during the build process.

6.3.2 Memory requirements for querying

When an analytics index has queries enabled, the index is loaded into RAM in an lsiapp.exe process. For indexes with millions of documents, this RAM requirement can be thousands of MB. The RAM requirement is dependent upon the number of unique terms in the dataset. Therefore, the range of RAM needed for an index to be enabled is as follows:

- (Number of searchable documents) * 5000 = High end of RAM required (bytes)
- (Number of searchable documents) * 400 = Low end of RAM required (bytes)

It's a best practice to Disable Queries on any analytics index that isn't used. This frees up RAM on the analytics server, and you can re-enable them instantly.

The configuration variable MaxAnalyticsIndexIdleDays assists in disabling unused indexes. This value controls the number of days that an analytics index can remain idle before it's automatically disabled by the Case Manager agent.

6.3.3 Server requirements for structured analytics

Structured analytics operations execute using the Java process on the analytics server. Ensuring that Java has enough RAM prevents restarts of the service or out of memory exceptions. You should also optimize the agent server to ensure that the export and import processes are as fast as possible. Ensure there are multiple Structured Analytics Worker agents. It is recommended to have, at minimum, 4 worker agents. Make sure that each worker agent has at least 1 GB RAM and 1 processor core available.

The structured analytics process also depends on CPU and I/O resources at various stages. Ensuring that the analytics server has multiple processors and fast I/O increases efficiency during structured analytics tasks.

6.3.4 Processing

When scaling Relativity Processing Worker machines horizontally, it's equally important to scale the storage if you want to continue to see linear improvements in performance. It doesn't help to put 20+ worker machines online if your storage system doesn't have the available IOPS and throughput to support them.

Processing is an application you use to ingest raw data directly into your workspace for eventual search and review without the need for an external tool. A processing job consists of two parts, file discovery, and file publishing.

6.3.4.1 Discovery phase

During the discovering phase, the Processing engine begins to ingest the files you specify in a processing set, and then OCR and/or text extraction are performed on those files.

Observations / Recommendations:

- During this phase, the only real shared resource is the file share.
- Depending on the storage subsystem, the source directory for the processing engine probably shouldn't be the same as the file share hosting Relativity review.
  - Excessive read I/O from the file share hosting Relativity natives and images required for review may increase Relativity viewer response times.
- The same is true of the destination directory.
  - Excessive write I/O to the file share hosting Relativity natives and images required for review may increase Relativity viewer response times.
- For workspaces where you will use processing, ensure the default file share is on separate storage from the rest of the files. If this is not possible, thoroughly test the I/O capabilities of the storage sub-system. This ensures that these operations don't negatively impact Relativity review.

6.3.4.2 Publishing phase

At any point after the completion of file discovery, you can publish the files that have been discovered in the processing set. During the publishing phase, the processed documents are made available for review via the workspaces document tab.

Observations / Recommendations:

- There is almost no contention on the file share as expected during this phase.
- There is some SQL contention as data is inserted into the workspace database.
- SQL processor utilization increases during the publishing phase.
- The SQL TempDB experiences increased usage, but no worse than a standard Relativity data import via the Relativity Desktop Client.

6.4 Marvel

All Relativity users who implement Data Grid can receive Elasticsearch Marvel licenses. Use Marvel to view your cluster status in a simple, single pane overview; investigate highly detailed system metrics;

visualize cluster events and metrics together to analyze how changes affect performance; and access the REST API.

See Marvel for more information.

6.5 Head

Head is a plugin for Data Grid that displays the overall cluster health, the different indexes residing in that cluster and the shards of the indexes. Using this plugin, you can quickly check on the health of the cluster.

See Head for more information.

6.6 SQL Server table index fragmentation

This section outlines the fragmentation of the SQL table column index, and describes how to eliminate it to ensure optimal system performance.

6.6.1 SQL table indexes overview

SQL table column indexes help improve system performance by reducing the amount of I/O (disk read and write) operations necessary to retrieve data.

These indexes speed up queries by providing quick access to data without the need for full table scans. A full table scan occurs when SQL has to read every row in a table to obtain the information it's querying. Each Relativity database contains many indexes to help ensure optimal performance. It's important to maintain these indexes throughout the life of a workspace.

6.6.2 Index fragmentation

Fragmentation occurs through the process of data modifications (INSERT, UPDATE, and DELETE statements) made against the table and to the indexes defined on the table. Because these modifications aren't ordinarily distributed equally among the rows of the table and indexes, the fullness of each page can vary over time. For queries that scan part or all of the indexes of a table, this kind of fragmentation can cause additional page reads and slow performance.

Note: This information was provided by the following Microsoft article: http://msdn.microsoft.com/en-us/library/ms188917.aspx.

6.6.3 Effects of fragmentation

If heavy fragmentation exists, performance is degraded and Relativity responds slowly, resulting in timeouts across various parts of the application. This includes searches, views, updates, deletes, data imports/exports, etc. Often a highly fragmented index can be worse than having no index at all.

6.6.4 Identifying fragmentation

A system admin can use the GetIndexFragLevels Relativity Script to determine index fragmentation for a database. The script identifies all indexes in a database with an average fragmentation over 10 percent and page count above 50. This is available in the Relativity Script Library of your environment.

This query places a significant strain on the SQL disks as it gathers current fragmentation levels. Don't execute this script on large workspaces while users are in the workspace. If the report returns nothing, no indexes are heavily fragmented in that workspace.

6.6.5 Removing fragmentation

We designed a smart script named IndexOptimize that eliminates index fragmentation in all Relativity workspace databases. IndexOptimize is available for download in the Relativity Community. Details are included in the comments section of the script.

Note: For instructions on setting up this SQL job, see SQL table index management on page 51.

Alternatively, you may wish to create a custom job that meets your specific needs.

We recommend scheduling **IndexOptimize** to run nightly.

7 Managing your Relativity environment

To effectively manage Relativity, follow these recommendations for backing up and maintaining your environment.

7.1 Workspace management and maintenance

Large workspaces with more than 500,000 records often require that system or database admins place additional indexes on any necessary fields in the SQL database. This ensures optimal workspace performance for the review team. For more information on SQL indexes, see SQL table index management on page 51.

For example, if a document list view includes a sort condition on an un-indexed fixed-length text field in a multimillion record workspace, each time a user selects that view, the executed query sorts on millions of rows. This can take extended time to complete. Each time a user selects this view it consumes significant server resources. Placing an index on this database's eddsdbo.Document table column can help reduce the query time.

You can use the provided generic SQL script to create indexes. Execute the following command to place a non-clustered index on a field:

```
CREATE NONCLUSTERED INDEX [IX_FieldName] ON [EDDSDBO].[Document] ([FieldName] ASC)
```

- Execute within the context of the database where you want to add the index.

- Replace the two instances of FieldName with the name of the field you would you want to place an index on (IX_ is just a naming convention). Consider including your initials here for identification purposes.

- Remove any spaces or special characters in the name in the script.

- Execute this script after-hours. This script locks the table for writes during the index build.

In addition to Fixed-Length Text fields, you can apply indexes to Yes/No (Boolean), Decimal, Whole Number, or Date fields on the document table of a workspace database. System admins should identify which fields are regularly queried, sorted, and filtered on in large databases for index creation.

Note: Any manually created indexes carry over during workspace creation. Relativity performs a full backup restore operation and doesn't contain any logic to delete indexes.

Understanding when and where to create additional non-clustered indexes improves with experience. Avoid indexing everything, because a large number of additional indexes can slow down insert and update operations. This can also increase the size of the database which increases maintenance, backup, and replication times.

Read indexes from the SQL Server more than you write indexes. SQL Dynamic Management Views collect and store index usage information. There are scripts available to query this data for reporting purposes. Restarting SQL Server purges data stored in the DMVs, so only analyze this information when the SQL instance has been online for at least a few days or weeks.

7.1.1 Analysis of long running queries

Long running queries take more than a few seconds to complete. This can include document list views, advanced searches, and other areas of the platform.

- Use the History tab in any workspace to identify long running queries. Include the Execution Time field in a History tab view. This helps to identify queries that take an extended amount of time to process.

- Use the History tab to view the actual SQL executed for every query. Relativity stores the information in the History tab in the eddsdbo.AuditRecord table for each workspace.

- Determine if you can optimize these queries to run more efficiently. For example, avoid nested or combined searches when possible. Avoid "Is Like" statements and simplify the criteria wherever possible.

- Use the estimated execution plan in SQL Server Management Studio to look for suggested indexes. You can also use the SQL Server Database Tuning Advisor for this purpose. If you use this tool, be mindful of the overhead involved as it analyzes the workload in order to make recommendations. If you can't make troublesome queries more efficient through Relativity, you can analyze them using the estimated execution plan or the SQL Server Database Tuning Advisor for suggestions. Apply any suggested indexes and statistics after hours, if necessary.

There are other SQL tools available to help identify long running queries, resource usage and overuse, and database locking issues. These tools are affordable and can assist in SQL Server performance troubleshooting. You can also use SQL Server Activity Monitor, Profiler, and Extended Events for monitoring purposes. Use caution and follow Microsoft best practices when using these tools, as they can be resource-intensive.

7.1.2 Full-text index management

In Relativity, the keyword search function queries the full-text index of a workspace's database.

You can query the health of a full text index in SQL Server. The following query returns the number of fragments in a full-text index. Execute it on the database in question.

```
SELECT OBJECT_NAME(table_id) AS [Table Name], fragment_id, fragment_object_id, [status], data_size, row_
count
FROM sys.fulltext_index_fragments WITH (NOLOCK)
```

For more information on full-text index fragments, see http://msdn.microsoft.com/en-us/library/cc280700.aspx.

If you experience slow keyword search performance, you may need to rebuild the full-text index for that database. You can determine this in SQL Server by querying the amount of FT fragments. Microsoft states that performance suffers significantly if an index has more than 100 fragments. We suggest rebuilding or merging indexes with more than 10 fragments.

To rebuild the FT Catalog for a workspace:

1. Navigate to that database in SQL MGMT Studio.

2. Expand the **Storage** > **Full Text Catalogs** folders, and then right-click **FT Catalog**.

3. Select the option to **Rebuild**.

During a full-text index rebuild, the workspace is still accessible. However, any keyword or "Contains" searches return incomplete results as the index populates. You should perform this task after hours. Select **Properties** for the index to view build status and progress.

You could optionally take a full backup of a database, restore it to the same server, and rebuild the full-text index for this restored database. Next, run an identical "Contains" query across each to measure any performance gain from a rebuild. You can also use this to gauge how long it takes to rebuild in the production database and schedule any necessary outage.

You can also use a smart script for managing full-text indexes during a scheduled maintenance window. Contact support@relativity.com to obtain this script and get more information.

> **Note:** For more information on full-text search internals and enhancements, see http://msdn.microsoft.com/en-us/library/cc721269.aspx.

7.1.3 Audit record table

Relativity stores all history for a workspace in the AuditRecord table for that database. The data in this table appears in the History tab for that workspace. Relativity audits all activity. An active workspace with 100 users can generate over ten million audits per month. This table could grow in size by hundreds of millions of rows in some scenarios.

If this table is very large, reducing its size can result in faster backup and index rebuild times.

Unfortunately, organizations often require all audit history be readily available to the review team. In some cases, it may be possible to delete some of the existing audit entries to reduce size and reclaim space.

Consider the following options to reduce the table size:

- Backup the database, backup the AuditRecord table, and then truncate the table (delete all existing audit history).

- Backup the database, backup the AuditRecord table to a certain date, and then delete everything from the table with a time stamp before that date.

If it's not possible to delete any audit entries, there are other options available to help assist with the management of this large table. See the Database Partitioning for Archiving Audit Records in the Relativity Community.

Please contact support@relativity.com for assistance with this operation.

7.2 SQL backups

You can set up all tasks mentioned in this document via the Maintenance Plan Wizard in Microsoft SQL Server Management Studio. Other third-party backup tools are available that are easier to manage, have a high compression ratio, and may take less time to complete.

Backup the following to ensure a full recovery of the system, if that becomes necessary:

- **Relativity Workspace Databases (.mdf, .ndf)** - all relevant databases are of the format EDDS#######

- **Relativity Workspace Database Logs (.ldf)** - all relevant database logs are of the format EDDS#######

- **Relativity System Databases (.mdf, .ndf)** - EDDS

- **Relativity System Database Logs (.ldf)** - EDDS

- **Relativity Processing and Native Imaging Databases** – INV#######, Invariant, RelativityImaging

- **Relativity Processing and Native Imaging Logs** - INV#######, Invariant, RelativityImaging

- **System Databases and Logs** - master, model, msdb (there is no need to back up tempdb)

The Full-Text Engine in SQL Server 2008 and later is fully integrated into the database, and full-text indexes are stored within database files (.ndf), rather than externally in the file system. Back up these files as well if they're in a separate file group.

7.2.1 Full backups

A **Full Backup** backs up the entire database. This includes the transaction log so you can recover the full backup.

Creating a full backup is a single operation, usually scheduled to occur at regular intervals. We recommend running a full backup nightly or weekly. Writing to fast storage, maintaining log file sizes, and including backup file compression can help reduce full backup execution time.

Although not required, verifying a backup is a useful practice. Verifying a backup checks that the backup is physically intact, ensures that all the files in the backup are readable and can be restored, and confirms that you can restore your backup in the event you need to use it.
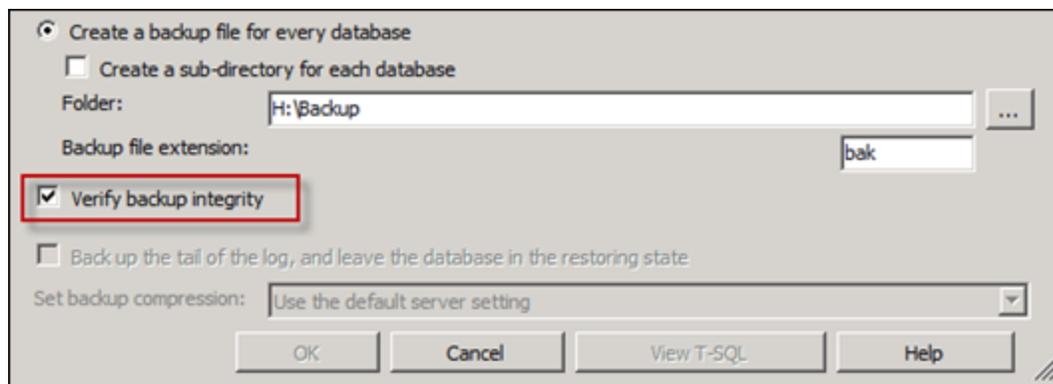
7.2.2 Differential backups

A differential backup contains all data that has changed since the last full backup. This is known as the base. Differential backups are quicker and smaller than full backups because they contain only what has changed since the previous backup.

You can only restore a differential backup after restoring a full backup. Differential backups provide no point-in-time restore capabilities.

7.2.3 Transaction log backups

Transaction log backups provide point-in-time restores or point-of-failure restores of databases.

Another advantage of transaction log backups is that SQL Server automatically truncates the inactive portion of the transaction log after it finishes backing up the transaction log. This inactive portion isn't used during the recovery process and contains completed transactions. The active portion of the transaction

log contains transactions that are still running and have not yet completed. SQL Server reuses this truncated, inactive space in the transaction log. This prevents the transaction from growing and using more space.

In our hosted instance, we make full backups nightly and transaction log backups hourly during the day. Other partners may only take full backups on the weekends and differentials nightly, along with transaction log backups during the day. You may have to adjust your backup schedules to accommodate different review schedules.

For additional best practices on maintaining SQL Database Log (ldf) Files, see the Managing Relativity SQL Log Files document in the Relativity Community.

7.3 SQL recovery models

Recovery models control transaction log maintenance. There are three different types of recovery models you can set at the database level.

The default databases are set to use the Full recovery model. This setting is carried over to any new databases because Relativity performs a full backup and restore operation during workspace creation.

Relativity supports the Simple recovery model for all workspaces. This decision would be entirely dependent on the disaster recovery requirements of your team. The considerations with making this change are detailed in the following MSDN article: http://msdn.microsoft.com/en-us/library/ms189275.aspx.

7.4 Relativity data backups

In addition to the SQL data, we recommend backing up the following items on a regular basis:

- **dtSearch and Analytics Index Shares** - these can be rebuilt, but the process may take days depending on the amount of data.

- **Relativity Web Server Install Directories** - back up the the IIS Relativity virtual directories to ensure you can recover any event handlers, custom images, etc.

- **Native and Image File Shares** - larger Relativity installations can potentially have millions of these files. The storage unit housing these files most likely includes several layers of redundancy. You must decide what backup solution works best for your team.

7.5 Check Database Integrity task

The Check Database Integrity task checks the allocation and structural integrity of all the objects in the specified database. This task executes the DBCC CHECKDB Transact-SQL statement. This ensures that any integrity problems within the database are reported to a system admin to address. For details, see http://msdn.microsoft.com/en-us/library/ms176064.aspx.

We recommend scheduling this task to run weekly. Ideally, you should run this task nightly. The earlier you find any database corruption, the quicker you can address it and reduce the risk of data loss. Unfortunately, this task can be very time consuming and resource intensive on larger databases.

Database corruption is usually caused by the underlying storage. It's important to schedule and monitor this task. If the task completes successfully, then there's no corruption in the analyzed databases.

7.6 SQL table index management

It's important to eliminate table index fragmentation in all Relativity databases. For more information, see SQL Server table index fragmentation on page 44.

We developed the IndexOptimize smart script to eliminate index fragmentation and update statistics in all Relativity workspace databases. You can download IndexOptimize in the Relativity Community. Details are included in the comments section of the script. We recommend scheduling this task to run nightly.

Note: See the Index Optimize Maintenance Plan Setup Instructions guide for assistance with setting up and scheduling Index Optimize along with other SQL maintenance plans. This guide is included in the .zip file when you download the IndexOptimize tool from the Relativity Community. If you need a Relativity Community account or assistance with setting up your maintenance plan, contact support@relativity.com.

7.6.1 Updating statistics

The SQL Query Analyzer uses statistics to choose the best path for obtaining data in order to boost query performance.

This task updates query optimization statistics on a table or indexed view. The query optimizer already updates statistics as necessary to improve query plans by default. The Index Optimize maintenance plan includes statistics maintenance as well. The default settings are ideal for most instances, but can be adjusted if necessary. You can also improve query performance by using UPDATE STATISTICS or the stored procedure sp_updatestats to update statistics more frequently than the default updates. For details, see http://msdn.microsoft.com/en-us/library/ms187348.aspx.

Consider updating statistics after performing maintenance procedures that change the distribution of data, such as truncating tables or performing bulk inserts of a large percentage of rows, such as data imports. This can avoid future delays in query processing while queries wait for automatic statistics updates.

Note: For more information on updating statistics to improve query performance, see http://msdn.microsoft.com/en-us/library/ms190397.aspx.

7.7 Database log management

The database transaction log file (.ldf) stores details of all modifications performed on the SQL Server database and the details of the transactions that performed each modification.

7.7.1 Size management

SQL Server automatically marks the inactive portion of the transaction log for reuse after it finishes backing up the transaction log. This inactive portion contains completed transactions and isn't used during the recovery process. The active portion of the transaction log contains transactions that are still running and have not yet completed. SQL Server reuses this truncated, inactive space in the transaction log. This prevents the transaction from growing and using more space.

We recommend scheduling transaction log backups to occur every hour or less during the day to keep these file sizes in check.

Although you can manually truncate the transaction log, we strongly advise against doing this because it breaks the log backup chain. The database isn't protected from media failure until you create a full database backup. Use manual log truncation only in very special circumstances. You should create a full database backup as soon as it's practical for you. You can truncate and shrink the transaction log file for a database with the following SQL script.

```
USE EDDS#######
GO
-- Truncate the log by changing the database recovery model to SIMPLE.
ALTER DATABASE EDDS#######
SET RECOVERY SIMPLE;
GO
-- Shrink the truncated log file to 1 GB.
--Specify the correct name of the file
DBCC SHRINKFILE (EDDS#######_Log, 1024)
-- Reset the database recovery model.
ALTER DATABASE EDDS#######
SET RECOVERY FULL;
GO
```

Transaction logs can present problems because they are often forgotten about until an issue occurs. The log continues to grow as users perform operations within the database. As the log grows, available disk space decreases. Unless routine action is taken to prevent this, the transaction log eventually consumes all available space. If the log is configured to grow indefinitely (which is the default), it consumes all available physical disk space where it's stored. Either scenario causes the database to stop functioning.

Regular backups of the transaction log help prevent it from consuming all of the disk space.

For additional best practices on maintaining SQL Database Log (ldf) Files, see the Managing Relativity SQL Log Files document in the Relativity Community.

7.7.2 Virtual log file (VLF) management

If you don't pre-allocate database transaction log files, they can become internally fragmented. These files can also become internally fragmented if there are many auto-growth operations.

SQL Server 2012 automatically checks for the amount of VLFs in database log files. It then writes a warning event for database logs with an excessive amount.

7.8 Shrink Database task

The Shrink Database task reduces the size of SQL Server database and log files but this does come with a cost to performance. For more information, review the following article Why you should not shrink your data files. Relativity doesn't recommend the shrinking of Relativity workspace databases due to the performance problems associated with doing so.

7.8.1 Best practices

However, if you still want to shrink a database please consider the following information before running the operation:

- A shrink operation is most effective after an operation that creates lots of unused space, such as an operation to truncate or drop a table. An example would be after a large deletion of documents in a Relativity Case that you don't expect additional loads to happen in the near future. However, be sure to turn off snapshot auditing during the deletion otherwise the audit table will contain a lot of the data.

- Most databases require some free space to be available for regular day-to-day operations. If you shrink a database repeatedly and notice that the database size grows again, this indicates that the space that was shrunk is required for regular operations. In these workspaces, repeatedly shrinking the database is not useful.

- A shrink operation doesn't preserve the fragmentation state of indexes in the database and generally increases fragmentation to a degree. This is another reason not to repeatedly shrink the database.

Note: For more information on shrinking the database, see http://msdn.microsoft.com/en-us/library/ms190488.aspx.

7.9 Job email notification alerts

Job email notification alerts are designed to notify system admins when a SQL job has completed and whether it was successful. It's important to configure these email alerts.

Set up SQL Database Mail on all Relativity SQL Servers and add these notification alerts to all scheduled Relativity maintenance tasks. Configure the alerts to email system admins when jobs succeed or fail. Whenever possible, set the operator's email address to a distribution list rather than an individual email.

7.10 Relativity applications

You can extend the functionality and features available in Relativity by installing custom applications to workspaces. You can also add them to the application library. You also have the option to uninstall these applications when you no longer need them in your environment. To ensure optimum performance of Relativity, uninstall applications only during off-hours when users aren't actively reviewing documents in any workspaces.

For more information, see Application Deployment System on the Relativity Server2021 Developers site.

8 Expanding your Relativity environment

Use the following additional features and tools to expand your Relativity environment.

8.1 Agents

Agents are process managers that run in the background of Relativity. Use the **Agents** tab to see which agents are running in your environment and the status of those agents. Adding multiple instances of certain agents can increase the speed of imaging, productions, OCR jobs, and index builds.

In general, use one processor core and 1 GB of memory for each additional agent. You can house these agents on multiple servers in the Relativity environment. For example, you may want to incorporate another agent server to increase the speed of productions. Allocate eight processor cores and 8 GB of memory to house eight additional Branding Manager agents.

Starting in Relativity 7.5, system admins can adjust agent configurations through the Relativity interface. It's important for system admins to understand agent best practices before adjusting agent configurations. For more information, see the Agents guide.

8.2 Analytics servers

Relativity supports multiple Analytics servers in order to allow jobs to run concurrently and to allocate resources among different workspaces.

When implementing a multi-server analytics environment, take the following considerations into account:

- What is the average dataset for analytics?

- What is the largest potential dataset for analytics?

- What type of jobs will be running on the server?

While the minimum system requirements for analytics are specified as 32 GB of RAM, this server has limitations. A server of this size could not build an index with more than five million training documents. Depending on the number of indexes enabled on the server, the limitation may be even smaller than this.

Beginning in Relativity 8.0, there are two main feature sets in Analytics—structured analytics and conceptual analytics indexing. These feature sets have differing requirements when considering the RAM requirements for Java.

For this reason, if heavy structured analytics usage is anticipated, it is highly recommended to have an analytics server dedicated to structured analytics.

Note: Never install any Relativity agents on an Analytics server.

To implement this configuration, install the Analytics component on an additional server. For instructions on installing Analytics, see Upgrading or installing Analytics in the Upgrade guide. After installing the Analytics component to the server, add the server to the Servers tab at Home. For instructions on adding analytic servers, see Adding an Analytics Server in the Admin guide. When adding the server in Relativity, select the analytics server type:

- structure analytics

- conceptual analytics indexing

8.3 Dedicated web servers

You can install additional web servers for specific Relativity roles, such as Relativity Desktop Client (RDC) imports or exports.

8.3.1 Relativity Desktop Client import or export

You can improve local data import and export performance by using a web server separate from the primary review web server. Doing so also reduces the load on the primary review web server.

To do this, install the web components of Relativity onto another web server that meets our web server requirement specifications. Next, update the WebService URL in the settings of the Relativity Desktop Client to reflect the new server name, **http://servername/RelativityWebAPI/**.
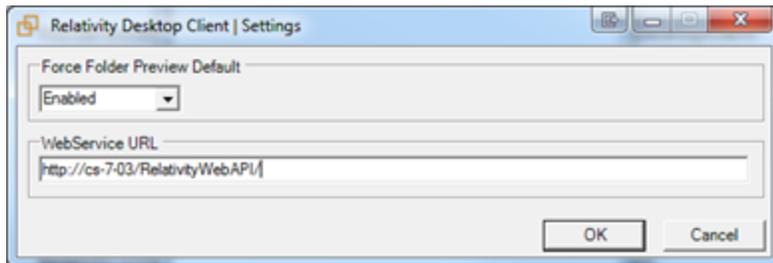
Note: You must install all web components, but you can disable the Relativity, RelativityREST, and RelativityServices application pools in IIS.

8.3.2 Physical memory on the SQL Server(s)

Install additional memory on the Relativity SQL Server(s) for an overall performance gain. The more memory you install, the less SQL will have to access disks. Disk access is slower.

The majority of SQL Server memory is used for the following processes:

- Each time SQL Server reads from disk, it commits those pages to memory so that future requests for the same data are cached in the Buffer Pool and readily available.

- The Procedure Cache stores the most optimal Execution Plans in memory so that they don't need to be recompiled each time, which can be processor-intensive.

The more information that you can store in memory, the better. The following table specifies the maximum memory supported for each edition of SQL Server.

SQL Server Edition	Maximum Memory Supported
Enterprise	Operating system maximum
Standard	64 GB in SQL Server 2012 128 GB in SQL Server 2014 and SQL Server 2016

After installing additional memory, be sure to set the **SQL Server Max Server Memory** server configuration option. For more information, see Configuring SQL Server on page 17.

We suggest that you make enough memory available to cache the eddsdbo.Document table of your largest, most active workspace(s) in RAM. If you don't host any particularly large workspace databases but host many, you should have enough memory to cache the three largest workspaces combined. SQL Server page life expectancy is a good metric to monitor in order to determine if your SQL Server has enough memory.

8.4 SQL Server failover clustering

SQL Server failover clustering ensures SQL Server availability and provides protection in the event of hardware failure. Failover to another node is quick and provides service availability, but it doesn't provide data redundancy. Instead, data protection is usually provided at the storage level.

(Source: http://www.mssqltips.com/tipprint.asp?tip=1882.)
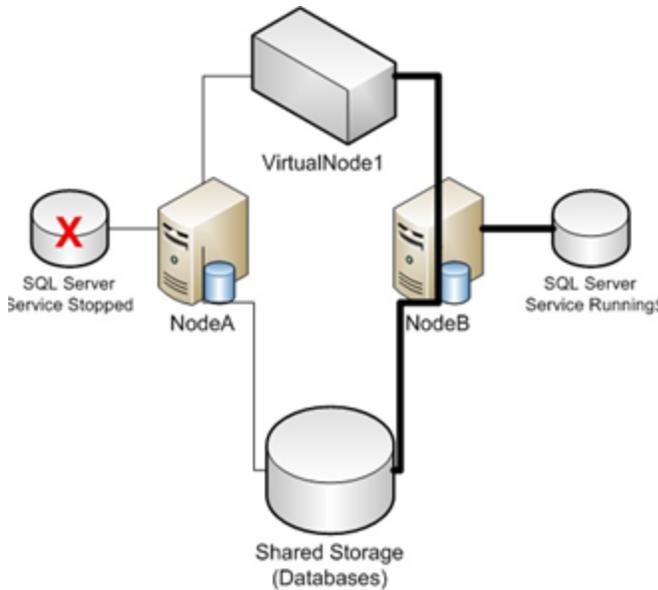
A SQL Server cluster appears on the network as a single SQL Server instance on a single computer. Internally, only one of the nodes owns the cluster resource group at a time, serving all the client requests for that failover cluster instance.

In case of a failure (hardware failures, operating system failures, and application or service failures) or a planned upgrade, group ownership is moved to another node in the failover cluster. This process is called failover.

By leveraging the Windows Server Failover Cluster functionality, SQL Server failover cluster provides high availability through redundancy at the instance level.

(Source: http://msdn.microsoft.com/en-us/library/ms189134.aspx.)

SQL Server clustering is only supported in Enterprise editions of Windows Server. The number of nodes supported for failover clustering depends on the operating system and the SQL Server Edition.

8.5 Distributed Relativity SQL Servers

Scale out the existing Relativity environment by incorporating additional SQL Servers. This helps distribute the load and you can store larger, more demanding databases on their own dedicated set of resources. Any number of Relativity servers can be included in an environment.

Each additional Relativity SQL Server is referred to as a "Distributed" SQL Server, unlike the Master SQL Server, which houses the EDDS database. There's only one EDDS database in a Relativity environment.

You can create new workspaces on any distributed SQL Server. You can also migrate existing workspaces between servers as needed.

Note: For more information, see The Relativity Installation Guide.

8.6 Data Grid nodes

As the business and data volumes keep growing, so will the Data Grid indexes. Indexes have a max of data and a max amount of shard that they can hold per Tier level. Scaling Data Grid data nodes is similar to scaling additional servers in Relativity in the sense that we add more and more Data Grid nodes to share the load of the data that is being placed on them and the queries that each data node has to service. Adding data nodes to the cluster does not require the cluster to be taken offline. When a data node comes online, the software automatically rebalances the indexes and shards among the data nodes to the most optimal configuration.

Client nodes are scaled in the same way. The more users and cases that are placed in the Relativity Data Grid cluster, the more requests will come into Data Grid. With more requests coming in, the more client nodes are needed to service those requests. The current Data Grid System Requirements call for 1 client node for a Tier 1 environment, 2 client nodes for a Tier 2 environment and 3 client nodes for a Tier 3 environment.

Monitoring system performance with Marvel is a good way to assess the need to scale in the environment. Marvel captures trending data and shows you how the environment has been performing over time with data and users increasing.

8.7 Web load balancing

You can use multiple web roles to distribute Relativity user load.

8.7.1 Microsoft network load balancing

Relativity supports different methods of web load balancing and the ability to distribute user sessions across multiple servers, which distributes the server load and allows for redundancy. Currently, Relativity supports Windows Load Balancer, set to Single affinity mode. Relativity doesn't support an affinity of None at this time.

Single affinity directs a specific IP address to the same server every time until the NLB Cluster is broken, a new server is added to the cluster, or a server is turned off. If you have multiple users behind a firewall, all of the users are directed to a specific server, and they stay there until the cluster is broken. This doesn't provide true NLB support, but it's required because the viewer uses an authentication token to create a second connection on the web server.

Note: Contact support@relativity.com to obtain NLB setup documentation.

Note: If you are using cookie-persisted load balancing configurations for Relativity 9.5.162.111 and above, you must update the **WebClientRequiredCookies** instance setting to include the name of the load balancer cookie for the ActiveX viewer.

8.7.2 Relativity user load balancing

You can use Relativity's Enable User Load Balancing feature to equally distribute user loads across all web servers that have Enable User Load Balancing set to Yes. When a user connects to the login page, the platform looks at the user status table to determine how many users are logged in to each web server. The user will then be re-directed to the web server with the lowest number of logged in users.

This requires setting up multiple URLs in addition to having multiple web servers or VMs. For more information, see the Admin Guide.

> **Note:** This feature only load balances users logging in to the system, not Relativity Desktop Client sessions.

8.8 Terminal services

To accommodate users with poor network connectivity, consider providing some type of a terminal services option. We suggest having Terminal Services solution available as a backup. The hardware required for this server role is dependent on the number of users it needs to support.

Deploying a program on a terminal server instead of on each device provides the following benefits:

- You can quickly deploy Windows-based programs to computing devices across an enterprise. Terminal Services is especially useful for programs that are frequently updated, infrequently used, or difficult to manage.

- It can significantly reduce the network bandwidth required to access remote applications.

- It improves user productivity. Users can access programs running on a terminal server from devices such as home computers, kiosks, low-powered hardware, and operating systems other than Windows.

- It provides better program performance for branch office workers who need access to centralized data stores. Data-intensive programs sometimes don't have client/server protocols that are optimized for low-speed connections. Programs of this kind frequently perform better over a Terminal Services connection than over a typical wide area network (WAN).

For this and more information on Terminal Services, see http://technet.microsoft.com/en-us/library/cc755053(WS.10).aspx.

8.9 Bandwidth tester

The Relativity bandwidth tester is designed to test the capacity of a network connection. You can execute this application at any time. It's valuable as a troubleshooting tool for users experiencing viewer issues due to a lack of data rate support from a network connection.

To launch the bandwidth tester, click the **Launch Bandwidth Tester** icon in the bottom right corner of the viewer, to the left of the Layout/Related Items pane.

For more information on the bandwidth tester, see the Admin Guide.

8.10 User notifications

There are several notifications available to help system admins better manage their Relativity environments.

8.10.1 Disclaimer Message

When you want users to agree to a disclaimer message when using Relativity, you can create a custom message and an agreement button for users when they first log in to the Relativity environment. This type of message is useful to ensure that users acknowledge Relativity terms of use or the confidential nature of Relativity content.

To customize a disclaimer on a Relativity login page, use the following steps:

1. Navigate to the web server(s) at C:\Program Files\Relativity Corporation\Relativity\EDDS.

2. Locate the text file, Disclaimer.txt, and update it with your disclaimer text. You must use plain text only; HTML will not display here.

3. Update the DisplayDisclaimer instance setting value to true to display the disclaimer message.

8.10.2 Message of the Day (MotD)

The Message of the Day (MotD) is a message displayed to all users when they log in to Relativity. MotD is most commonly used to inform users of planned system maintenance.

To activate or change the message of the day, navigate to the Instance Details tab from Home.

8.10.3 User status

The user status page displays a list of users currently logged into the system. To access the user status page, navigate to the **User Status** tab from Home.

You can also send messages to any logged in users. This is helpful if you require emergency downtime during a review. System admins can also force log out users from the system on this page.

8.10.4 Default workspace tabs

When a user enters a workspace within Relativity, they are usually taken to the **Documents** tab. You can change this default setting in the **Tabs** tab of any workspace.

Set the **Is Default** property to Yes for the tab that you want to designate the default tab in a workspace. If a user doesn't have access to the Default tab, he or she will be directed to the Documents tab.

Modifying the default tab may be helpful in informing users of any upcoming workspace-level maintenance. Additionally, system admins can direct users to custom pages that contain links to instructional items, important information about the matter, or an overview of workspace review progress.

Proprietary Rights

This documentation ("**Documentation**") and the software to which it relates ("**Software**") belongs to Relativity ODA LLC and/or Relativity's third party software vendors. Relativity grants written license agreements which contain restrictions. All parties accessing the Documentation or Software must: respect proprietary rights of Relativity and third parties; comply with your organization's license agreement, including but not limited to license restrictions on use, copying, modifications, reverse engineering, and derivative products; and refrain from any misuse or misappropriation of this Documentation or Software in whole or in part. The Software and Documentation is protected by the **Copyright Act of 1976**, as amended, and the Software code is protected by the **Illinois Trade Secrets Act**. Violations can involve substantial civil liabilities, exemplary damages, and criminal penalties, including fines and possible imprisonment.