Relativity®

Backup and Data Management Best Practices

September 17, 2019 | Version 10.1.290.1

For the most recent version of this document, visit our **documentation website**.

Table of Contents

1 Overview

This guide provides an overview of considerations for data recovery and developing a Relativity backup strategy. Often, backup strategies are inconsistent, poorly documented, and misunderstood. This misunderstanding stems from widespread expectations that backups must exist, must be readily available at a moment's notice, and must be restored immediately, regardless of size. When backup and Disaster Recovery (DR) practices are not documented, management believes that not only do they exist, but that they are good, consistent, and that they are current (possibly up to the minute). It is therefore critical to document your organizations SQL backup strategy and practices.

This guide assumes an understanding of the following technologies: Redundant arrays of independent/inexpensive disks (RAID), tape backup, error checking, hash algorithms, and general security strategies.

To deliver in-depth support for litigated matters, Relativity uses a very compartmentalized approach to database storage. Each individual workspace is set up with a dedicated database. For strategic performance reasons, these databases are often moved between servers. Relativity also provides processing functionality, which ingests native files into databases. With Relativity Processing installed, each workspace that uses processing has a "sister" store database. These sister store databases begin with the letters "INV" and exist on a separate database server.

Document long text fields and workspace audit history can optionally be stored in Relativity Data Grid rather than in SQL.

1.1 Protecting backed up data

After backing up data, some system admins won't worry about data maintenance. They make the assumption that, once "backed up," data remains consistent and complete. However, experienced system admins know that databases, and data stored on disk in general, can become corrupted just from sitting on a disk. Further risk is introduced when third-party data synchronization components and snapshotting technologies are used.

For this reason, you must adopt in-depth backup strategies for all business-critical data. All Relativity backups—especially those taken offline—should have multiple BAK copies in different locations, and they should be periodically checked for consistency. For SQL backup strategies, see Selecting a backup approach on page 8.

Ultimately, you should protect the data according to business demands. However, consistency checking puts additional strain on both infrastructure and personnel. You must scale these resources according to the acceptable level of data loss based on business requirements.

First, you must determine the mean time to failure (MTTF) of a system. Next, based on the business requirements, you need to determine the acceptable data loss tolerances. Once you fully understand these factors, you can implement the appropriate number of redundant disks to provide a first layer of defense against data loss.

1.1.1 First line defense

More disks mean greater redundancy. The ability to have data striped across disks improves redundancy. In RAID 1+0, every disk you add increases redundancy. With bit-level striping, bytes are striped across multiple disks. Storage redundancy in RAID is your first line of defense. When a hard drive starts to become corrupt, the storage controller can take it offline. Once it is replaced, rebuild its contents using the non-corrupt contents of other drives. This occurs automatically online.

A successful Data Loss Prevention (DLP) strategy leverages knowledge of MTTF against mean time to repair (MTTR). It has an understanding of disk striping in order to maximize reliability at the online storage layer. For more information on establishing backup maintenance procedures, see Selecting a backup approach on page 8.

Backups, or offline data, are another way to mitigate risk. An additional strength of the strategy comes from having multiple copies of a file and the ability to detect a failure of the data before it's lost. Other countermeasures, such as non-volatile RAM (NVRAM) cache, also help prevent data loss. Such solutions haven't become mainstream yet.

Microsoft SQL Server always keeps the most recent data in RAM and writes the data to disk as load permits. If SQL Server crashes and fails over to a clustered server, the data in RAM on the downed server can't be recovered. If necessary, assess and adopt NVRAM technology.

This guide outlines several approaches to backing up data and provides information on redundancy maintenance practices. Ultimately, data retention is a business decision. The cost of doing business, profit requirements, and the potential damage of data loss are all business concerns. This guide doesn't cover disaster recovery options.

Note: For disaster recovery (DR) options, see the Configuring Relativity after Disaster Recovery Failover document on the Relativity Community . There are many options for replication/mirroring of a site for failover in a Disaster Recovery situation. This document outlines the necessary steps to take after a failover in order to return Relativity to an operating state.

1.1.2 Second line defense

The second line of defense is nearline data storage. Data is nearline if it can be brought online quickly. For example, your most recent backup file saved on a SAN is "nearline" and immediately accessible . It should be free of corruption.

1.1.3 Third line defense

The third line of defense is offline data. Offline data can be removed from nearline data in both time and space. For large data, you can manually ship an offline data backup or move it over the Internet over a period of several days. The time differential depends on the following factors:

- Cost of storage
- The value to the business of maintaining large amounts of recent data, offline and far away
- Logistics

For highly mission-critical data, you must synchronize the data on a daily, if not hourly, basis. For example, a disaster recovery data center may be third-line data. To establish a Disaster Recovery (DR) site, you can use technologies such as log-shipping over high-speed Internet or mirroring. There also exist mechanisms with SAN and virtual technologies to keep data that's far away almost up to the minute. This is very expensive and requires tremendous expertise to setup and maintain. It will almost inevitably impact production environment performance.

2 Identifying data to back up

The following sections provide a comprehensive list of all possible data locations of relevant Relativity files (there may be other areas that relate to custom applications or ISV products). You should also consider these areas when creating a backup inventory.

2.1 Database files

A backup of a SQL 2012/2014/2016 database includes the Data file, the Log file, and any additional data files, such as the full-text index catalog. You should preserve and thoroughly document maintenance plans and other SQL configurations to help restore service after an outage. There are many configuration options available with the SQL backup command—understanding them is critical to properly maintaining backup continuity. Both Relativity workspace and processing databases should be backed up.

2.2 Relativity file repositories

Relativity file repositories are locations that Relativity "owns." That is, Relativity creates and deletes files from these locations when requested. The locations of Relativity file repositories are stored as choices in the EDDS Database CodeArtifact table in MS SQL. You can find these locations by running the following query:

```
SELECT [Name] FROM [EDDS].[eddsdbo].[Code] where CodeTypeID = '1000000'
```

2.3 File stores

Natives and images loaded with pointers are treated differently. Relativity reads files from file store locations but never deletes the locations, even if the document is deleted from Relativity.

Locating all file store locations is complicated. File path locations are stored in the File table and must be parsed out from any Relativity file repository paths. If you need assistance identifying these locations, contact support@relativity.com.

2.4 Relativity Analytics

The Relativity Analytics server contains critical information about some configurations and may also be the default location for Analytics indexes. Take care to ensure that this server can be completely restored.

2.5 Cache location

Cache location servers temporarily store natives, images, productions, and other file types the viewer uses. Backing up the cache location is optional but recommended as rebuilding the cache can take much time and planning.

2.6 Relativity Data Grid

Please see the Backing up Relativity Data Grid page for more information on backing up Relativity Data Grid.

2.7 Configurations

The following sections provide possible data locations for configuration information in Relativity.

2.7.1 Agents

Relativity retains agent configurations in the database. Backing up the EDDS database effectively preserves all agent configuration data.

2.7.2 Web

You can customize certain aspects of the Relativity web application. For this reason, you should back up the website files as needed. You can also customize certain configurations in IIS. IIS provides a way to export and save the IIS configuration.

2.7.3 Ancillary indexes (dtSearch, Analytics)

The configuration of the index specifies the locations of the dtSearch and Analytics indexes. Capture all folders and subfolders.

2.7.4 Full-text catalogs

The full-text catalog may reside within either the database .mdf file, a separate .ndf file, or a mixture of both. The distribution location of the full-text catalog may vary depending on the age of the instance as well as the original version of SQL.

3 Selecting a backup approach

You can implement one of the following approaches to backing up your data online:

- Active databases
- Inactive databases

3.1 Active databases

Relativity databases may experience a very high number of inserts and updates and can become corrupt at any time. An "active database" experiences moderate to heavy use. For this type of database, data loss would be catastrophic to business.

For your active databases, we recommend using one of the following backup strategies.

- Performing nightly full database backup below
- Performing weekly full database backup below
- Performing weekly full database backups with no log backups on the next page

3.1.1 Performing nightly full database backup

Follow these steps to perform nightly full database backups with log backups for point-in-time recovery.

Data file:

1. Perform a full backup of the database nightly.
2. Mirror to remote disaster recovery site. (Mirror can mean log shipping, SAN snapshots, or replication. Follow the established strategies of your business.)
3. Restore nightly backup to inexpensive equipment each day.
4. Run **DBCC CheckDB** each day or as often as possible, and meet the business requirements for data loss prevention.
5. Complete the **DBCC** before the next backup occurs.

Log file: Follow best practices for managing log files as outlined in the Managing Relativity Log Files guide. Understand the best approach for timely restoration to meet your recovery time objectives (RTO) and recovery point objectives (RPO).

Note: A Relativity database log file may occasionally experience a high amount of growth. If the log files fill the log drive, those workspaces become inaccessible. If this happens on the SQL Server that contains the EDDS database, the entire environment becomes inaccessible.

3.1.2 Performing weekly full database backup

Follow these steps to perform weekly full database backups with nightly differentials and log backups.

Data file:

1. Perform a full backup the database weekly.
2. Perform a differential backup of the database nightly.
3. Mirror to remote disaster recovery site.
4. Restore nightly backup to inexpensive equipment each day.
5. Run **DBCC CheckDB** each day.
6. Complete the **DBCC** before the next backup occurs.

Log file:

1. Follow best practices for managing log files as outlined in the Managing Relativity Log Files guide .
2. You should also back up log files as dictated by business needs for point-in-time recovery. This backup should overcome any possibility of filling the log drives.

Relativity may write a lot of data to the log files at times. If the business only requires a four-hour increment for point-in-time recovery, but the system could write enough data to the log files in four hours to fill the drives, then you must either run log backups more frequently, or increase the size of the drives. Be sure that you understand how to restore log files; document the procedure for restoring log files and practice doing it.

The following formula determines the higher bound constraint of frequency of log backups:

- If there is **x** GB of free space on the log drive, and the system can write data to the system at rate of **y** GB/hr, and **t** is time, then the frequency of log backups **F** is **F(t) = *b*(x/y)**. **b** is given by some integer less than one which is determined by the capability of the system to move data. If this number cannot be maintained at some value **b < 1**, then additional bandwidth from production disk storage to backup storage is required. This variable is then the ratio of synchronous read/write where it is some value less than one and represents actual demand on the system, not merely the capability of the system. In other words, the assumption is that the value of **b** will exist in some domain such that, during normal production activity, the ability of the system to read data from the log disks is not impeded by its write activity.

  For example, if during a normal hour of production 100 GB are written to the system and 100GB are read from the disk by production systems, and the remaining capacity is only an additional 50GB of sequential read data, then **b= 2** and this is not a value of **b< 2**. Log backups may not complete before the next round is scheduled, or the production system performance suffers because you're operating your system at the upper limit of what your system can handle.

Note: Whereas the lower limit of the frequency of log backups is controlled by the need of the business for point-in-time recovery , performing more log may be required by the need to prevent a drive from becoming full.

3.1.3 Performing weekly full database backups with no log backups

Follow these steps to perform weekly full database backups with nightly differentials and no log backups.

1. Perform a full backup of the database weekly.
2. Perform a differential backup of the database nightly.
3. Mirror to remote disaster recovery site.

4. Restore nightly backup to inexpensive equipment each day.
5. Run **DBCC CheckDB** each day.
6. Complete the **DBCC** before the next backup occurs.
7. Set recovery model to **SIMPLE**.

Log file:

In this configuration, set the recovery model of the database to **SIMPLE** and size the log files appropriately, as outlined in the Managing Relativity Log Files guide; you can find this guide on the Relativity Community. At a minimum, you should set the log files to the approximate size of the Document table. With this approach, you can only restore to the point in time of your full or differential backups, as no log backups are taken throughout the day.

Note: If you suspect excessive logging at any time, please report it to support@relativity.com to identify or determine a root cause.

3.2 Inactive databases

Inactive databases also require attention. You may not routinely back up inactive databases, but a stored backup may become corrupted over time for various reasons—such as head crashes, aging, wear in the mechanical storage devices, etc..

Data can become corrupted just sitting on a disk. This is called silent data corruption. No backup can prevent silent data corruption—you can only mitigate risk.

3.2.1 Preventing silent data corruption

The consequences of a silent data corruption may lay dormant for a long time. Many technologies have been implemented over the years to ensure data integrity during data transfer. Server memory uses Error Correcting Code (ECC), and Cyclic Redundancy Checks (CRCs) protect file transfers to an extent.

It's important to maintain the integrity of data at rest on disk systems that aren't accessed over a long period of time. Without a high degree of protection, data corruption can go unnoticed until it's too late.

For instance, a user attempting to access the database may receive the following error when running certain queries:

```
Error 605
Severity Level 21
Message Text
Attempt to fetch logical page %S_PGID in database '%.*ls' belongs to object '%.*ls', not
to object '%.*ls'.
```

Then, while running DBCC to try to repair it, the database administrator receives this error: "System table pre-checks: Object ID 7. Could not read and latch page (1:3523) with latch type SH. Check statement terminated due to unrepairable (sic) error."

After this occurs, the database administrator checks for backups, and hopefully recovery happens quickly and inexpensively.

Sometimes, the backup is also corrupt. Often times when this happens, there is no way to recover missing data (e.g., the database can't be repaired if complete tables have been destroyed.

When DBCC checks are not run regularly against backup files, a corruption such as the previous example may go unnoticed for weeks.

To prevent this data corruption for business-critical databases, the following steps should occur after completing every backup:

1. Restore the backup to inexpensive hardware running MS SQL Server.
2. Run **DBCC CheckDB**.
3. Create a hash of the database BAK file.

3.2.2 Example strategy

Follow these guidelines to prevent data corruption of inactive databases:

- Perform **DBCC CheckDB** against business-critical databases.
- Keep one week's work of backups.
- Run a file hash each night.
- Ensure that no file hash has changed. (A file that becomes corrupt will have a different hash.)

Freeware tools, such as ExactFile, include features that make it easy to test hash values by creating a digest of a directory of files. This digest can be tested and rebuilt daily or on a sensible schedule that meets business obligations.

4 Assessing risk and cost concerns

When it comes to backup verification, IT management has two competing concerns:

- Management of risk (including cost-of-failure and mean time to failure (MTTF) estimates provided to the business owner).
- Cost to business of maintaining backups.

If you know the cost of maintaining backups exceeds the value of the data being maintained, then you can relax retention rules. The business should adjust contractual obligations, real or assumed, as needed. If the loss of data significantly compromises the business or is potentially a business-ending event, then analysts should carefully weigh the cost of retention and retention maintenance against contractual obligations to the owner or customer.

4.1 Considerations

Consider the following when assessing these concerns:

- The cost of rebuilding the data set from scratch.
- The cost of silent failure prevention.
- The overall revenue indirectly and directly generated by the data.
- The profit margin of the data.
- The cost to business due to loss of reputation in the event of an unrecoverable failure.
- Any legal obligations regarding data retention and reasonable retention efforts.
- Insurance policies may cover loss of IP (good backup strategies consider the amount of coverage provided by insurance).

In addition, ask yourself these questions:

- If the data become irrecoverably corrupted, can it be rebuilt? Is the data irreplaceable, such as photographs or videos?
- Are you dealing with real-time decision data? How many hours of human decision data are invested in creating the data?
- Assume the worst-case scenario: You have corrupt, irreparable data that can't be rebuilt. What is cost to business of data loss?

4.2 Summary and conclusion

Establishing reasonable practices is key to any successful backup and data management effort. First, you must develop practices, document these practices, and ensure that IT personnel follow them. Then, you must understand MTTF and maintain a tolerance.

To achieve the highest level of reasonable data retention and silent failure prevention, consider retaining two copies of the data at two geographically distinct locations. Perform weekly MD5 hash checks on the data primary set and monthly checks of the secondary. You can automate these tests and run them in

addition to the initial consistency checks performed when the data was created. Developing sound backup and data retention procedures facilitates the safety of data, compliance with business needs for point-in-time recovery, and potentially regulatory compliance. Most database technology includes the ability to restore after a disaster. As such, the performance of the database depends on understanding the way in which logging and backups operate and impact performance. Improperly configuring backups can result in unanticipated outages and even loss of data if backups are corrupt.

For assistance with any of these configurations, contact support@relativity.com.

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.