Protecting Your Business

A Practical Guide to System Recovery using Rocket® jBASE® Transaction Journaling

Contents

Synopsis

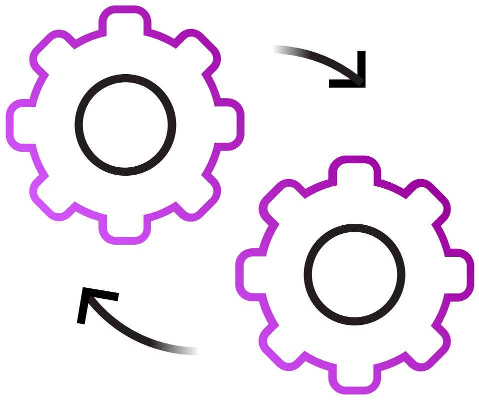
Downtime and data losses that were tolerated in the past are no longer acceptable in today's interconnected business world. Our systems must become more reliable, resilient, and recoverable.

This document:

- Breaks down the benefits of technologies that can cost-effectively deliver real value and ensure that your business survives system failures with minimal business impact.
- Explains why certain technologies, advertised to protect your business, are incapable of providing the protection you need.
- Debunks misconceptions about commonly used technologies and brings awareness to business owners, managers, and IT professionals about dangers lurking within your current systems.

Why is business continuity important?

The goal of business is to generate profit by providing a valuable product and/or service to customers. It's hard to imagine a time when businesses ran primarily on manual systems – 'simpler' times that no modern businessperson wants to go back to.



Technology brings efficiency to an organization by lowering operating cost while improving customer service levels. When it works, it is a thing of beauty. But when it fails, it can become the thing of nightmares.

Today, businesses are connected in ways never imagined even at the start of "computerization." EDI, eCommerce, data warehousing, email, customer portals, document management, and other advances have increasingly streamlined the business process. However, inter-connectedness means that a system failure has a more profound impact on a business.

Businesses have far less tolerance for system failure, downtime, and data loss, and yet many vendors still recommend and implement inadequate resiliency, based on practices from decades ago.

Cannot be made to fail

When building systems, there is an engineering credo that we should bear in mind which goes:

"Any fool can build a system that can be made to work, but it takes talent to design a system that cannot be made to fail."

It takes effort and money to create resilient systems. So, the benefit of resiliency must outweigh its costs, or you risk pursuing an extreme that goes beyond what is practical.

Over the years, technology has evolved, becoming more complex and powerful but also simpler and more affordable. Resiliency goals that were thought to be too expensive only a decade ago are now well within the reach of most small-to-medium sized businesses. And yet, most SMBs have failed to implement these capabilities to safeguard their work. In addition, some vendors of business solutions lack the understanding, skill, or motivation to properly guide their customers in this area.

More than in many other industries, there is a lot of misunderstanding in IT about how resiliency is achieved. Customers are often sold technologies that promise a benefit that the technology is incapable of delivering. High Availability, Disaster Recovery, and Failover technologies are, unfortunately, an area where customers are often sold a bill of goods only to discover, when it's too late, that this 'resiliency' has let them down.

← Back to contents

What are our goals?

For businesses, there should be two primary objectives in your business continuity strategy: Recovery Time Objective (RTO), and Recovery Point Objective (RPO).

Recovery time objective (RTO)

This objective is measured in the time required for processing to resume after a failure. A few decades ago, because computers and storage were expensive, the RTO solution was to purchase a computer support contract with a guaranteed response time. These contracts offered various response times such as same-day, 4-hour, or next-business-day. By the time the problem was encountered, reported, dispatched, worked, and resolved, a business could be down for a day or more. This was considered acceptable because the cost to purchase and manage a backup computer was simply too high.

Today, technology costs have plummeted, and it now makes sense for businesses to run on resilient technology. This can mean selection of servers with built-in resiliency features such as mirrored disks, redundant power supplies, multiple CPUs, and network cards, etc. While this level of protection is commonplace in today's hardware, most businesses will also protect itself with multiple servers configured in a way that if one fails, the other can take over processing in minutes or moments, rather than days. Cloud server vendors build in this level of resilience and this is one of the reasons why they have become so popular.

RTO may also consider the time to recover from failure events other than hardware failures. For example, if one of your administrators makes a mistake and damages or destroys critical data, can that data be recovered? How long would it take? This leads us into a discussion of our second objective, the Recovery Point Objective (RPO).

Recovery point objective (RPO)

RPO measures the amount of data you could potentially lose in a failure event. In the past, the standard RPO accepted by most SMB's was "up to the last good backup." In most cases, backups are performed once per day — so if a data loss happened moments before the next backup, the business loses the last 24 hours of data.

We have all heard the stories of system failures resulting in lost data. Sometimes all employees were subsequently tasked to re-enter lost transactions after the systems were restored to the last good backup. This is both stressful for the workforce and scrambles business records because it is nearly impossible to re-enter data in the original order. The result is mismatched transaction IDs (invoice numbers, etc.), plaguing the organization and breaking audit trails, which erodes customer confidence.

Today's businesses interconnectedness means that such data loss events cause a wide range of problems, like duplicate orders to vendors and inventory discrepancies, on account of manual inventory procedures while the systems were down.

In light of the technology of today having greater capacity and affordability, there is little reason for small-to-medium businesses to tolerate an RPO beyond moments or minutes.

← Back to contents

Myths of business continuity

There are a variety of technologies available for resiliency. But their capabilities are often misunderstood by business owners and managers, resulting in businesses buying the wrong tools.

MYTH #1: YOU HAVE A BACKUP

Do you really? The computer displayed “backup complete”, but does your backup afford you a comprehensive recovery? Here are a few examples of faulty backup strategies:

Are you backing up everything?

Years ago, your backup system was configured to back up the entire system. But your system has changed. You may have added disk, created new file systems, upgraded operating systems, created mount-points, and more. Some data is stored on users’ PCs or elsewhere in the organization. Have you audited your backup to verify that you have everything that you will need to recover? If you recover from the backup, will off-server data storage be in sync with the recovered data? For example, if you recover using your backup but your document management system contains documents that are referencing transactions that could not be recovered, what will happen?

Do you have a Point-In-Time backup?

Some backup strategies purport to allow a backup to run while the database is active. Advanced mechanisms can provide the backup process with an unchanging image of the database at a point-in-time; however, unless these mechanisms are implemented, your backup has limited usefulness as a recovery point.

Imagine it is midnight and your backup process kicks off. There are a few users on the system and services such as EDI, eCommerce, and others are running but the system is “pretty quiet,” compared to the middle of the business day. The multi-hour backup begins. Let’s say that the first module to be backed up is the inventory module with its data and the backup of inventory data finishes 15 minutes later. The backup continues for several hours and then reaches the accounting module. The accounting module’s data finishes its backup three hours later than the accounting data. The backup is now complete which contains inventory data from 12:15 a.m. and accounting data as of 3 p.m. If you were to recover using this backup, you can see that there is a possibility that the accounting transactions will not match the inventory levels. In the past, the solution was to block all activity by logging all users out during the backup, to ensure that the backup represented a single point in time data state.

In modern times, business systems have become so interconnected with the outside world that it is seldom possible to take a system off-line for hours to eliminate data state changes and to protect backup data integrity. Unless measures are implemented to allow for 24x7 access without causing changes to data state during the backup, recovering to the data state of the backup could lead to many downstream issues.

Do you verify your backups?

Nothing is worse than realizing you were misled by the message, “Backup Complete,” only to discover that you didn’t have enough disk space to store the backup, so the program just decided to tell you that your backup was complete. The only way to be sure that your backups are complete, and that they are readable and recoverable, is to verify them. Even if your backup software has a “verification” process, perform periodic restores and independently verify your restored data. This verification should include an audit of backup scope to ensure that every piece of critical data has been backed up.

Is your system quiet when you back up?

It is not possible to have a reliably coherent and recoverable backup of your system if the data you are backing up is in flight (in use). This applies to all databases and all systems. Disk or machine-level backups or snapshots, by themselves, should not be trusted. Special measures must be taken for any database to ensure that the backup is coherent and recoverable.

What is your retention policy?

If you have a good backup, how long will that backup be retained? If you do not know, you’re at risk. A purged backup is no backup. It is all too common to find backup systems configured to purge backup files after a set period. What is more alarming, is that some vendors have been observed reducing retention policies without notifying the business stakeholders. Often this happens because the customer may be running low on disk to store the backups. Other times, a customer may sign up for a cloud backup solution without understanding the retention policy. In some instances, reputable cloud vendors have been found to purge backups older than seven days while the business stakeholders thought they had unlimited backup retention. Thankfully, in these cases the disconnect was discovered before the customer needed their backup. Most businesses should have at least:

- 31 days of daily backups.
- 13 months of monthly backups.
- Several years of yearly backups depending. on the requirements of your industry.

In addition, businesses may run sensitive processes such as monthly/yearly closes. A business may want to create special 'before close' backups to serve as a recovery point should anything fail within the closing process itself. Of course, it goes without saying that any time one performs invasive maintenance on a system, it is prudent to make a complete, verified backup and to move it off the server.

Where are your backups?

It is generally accepted as a best practice that you should always archive your backup in three different places. Beyond the obvious redundancy that this strategy provides, it also shields you from risks such as:

- One of your destinations gave you a false indication that the backup safely landed there.
- The vendor keeping your backup suddenly goes out of business.
- One of your archive locations is breached, hacked, or crypto-locked.
- Connectivity to one archive location is lost at precisely the time it is needed.
- An archive is destroyed by accident or incident such as fire, hack, or crypto lock.

A common strategy is to keep one set of archives on-site, one set off-site, and one set in the cloud.

Who can recover?

Part of an effective backup and recovery strategy is to carefully document and update all aspects of the backup and recovery strategy. You must not leave it to one person who can become unavailable while holding access to your backups. A team of at least three persons should be nominated to manage backups. Every team member should possess the knowledge and information required to perform a perfect system recovery without interaction with any other team member.

Protecting credentials

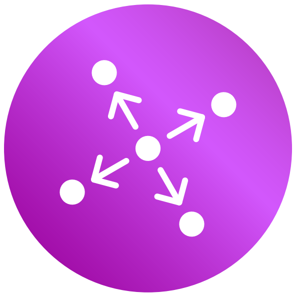
Remember that you should not place credentials such as login names and passwords into your procedural documents. These should be secured using proven password managers, usually with some sort of master password. And it is vital that this information also be carefully protected, placed in multiple locations, and distributed to at least three trusted persons within your organization.

Do you have confidence?

The only way to have confidence in your backup and recovery strategy is to practice it. You should schedule periodic mock recoveries using test servers. The practice should touch on all critical data and application functions. You should keep these practice procedures in full view of your IT team, and you should modify the documentation of the procedure each time it is practiced — noting the date, time, operator, and result of the test within the procedure document. If you notice something missing or in error within the document, you should update the procedure to ensure that when it is needed, the procedure is up-to-date, tested, and ready for use.

With all this said about backups, remember that backups are your fallback plan. They play a vital role, but they are not the tool of choice for achieving the low RTO and RPO objectives that are appropriate in modern business. For that, we recommend transaction journaling and replication to our Rocket jBASE customers, which is discussed later in this document.

MYTH #2: HARDWARE DATA REPLICATION IS THE ANSWER



There is a common misconception, even within IT circles, that features such as disk-replication, SAN snapshots, file-system replication, file replication, rsync, disk snapshots, virtual machine snapshots, and other technologies of this type will provide a safe copy of your running system. But this is simply not true.

The only way that these systems can ensure integrity of the replicated copy of the data is to either gain awareness of what the applications are doing to the data state, or to ensure that the data state is consistent in all respects before the replication has started — and to ensure that the data state cannot change while the replication is in process. To achieve this, the snapshot, replication, copy, or backup process would need to ensure that:

1. All data in memory is flushed to disk.
2. All writes to disk have been suspended so that we can know that the data state is not changing during a replication event.

In some cases, virtualization systems can have the ability to save both the entire machine and the content of its memory. In theory, this should provide a level of protection against physical file corruption but may introduce other complications to the recovery process and does not address the logical integrity of the data.

Most modern databases provide commands to suspend database updates. However, it is seldom possible to suspend updates during the business day at frequent enough intervals to achieve an acceptable RPO.

Physical versus logical integrity

Suspending database updates before and during a backup will ensure the physical integrity of the data being replicated. It cannot, however, ensure logical integrity of the transaction. For example, assume your system is running a batch update that involves an update to a customer record, a GL record, an order record, and four inventory records. The database suspend directive is issued just after the GL record is updated and before the order record is updated. In that situation, the data state will be physically intact (no corruption), but if you switch to the replicated image from that snapshot, the batch transaction would be missing its order and inventory updates, and the transaction would be considered logically incomplete.

Most databases have commands that can be added to applications and surround related database updates into a unit called a transaction. If your application is written that way, or can be modified in that way, then logical integrity is ensured because the series of related database updates are performed together either all, or none. If your application does not use defined transaction boundaries, then the database will consider each update to be its own transaction. This can lead to a logical corruption of transaction.

MYTH #3: HARDWARE DATA REPLICATION ALONE GIVES RPO

While there are plenty of salespeople and even IT professionals that might agree, this is simply not true. Hardware data replication systems are useful in certain contexts, but they all face the same challenges with data integrity. They cannot be aware how the application and its database update data so they cannot know how to properly choose the precise moment when it is safe to create the replica. To ensure that the replication system chooses the right moment, you must force a steady data stage by forcing your database to suspend its updates. For most databases and snapshot systems, this results in a considerable "pause" up to minutes in duration. This is obviously noticeable to users and may even cause the failure of processes. Because of this noticeable impact to users and processes, businesses usually limit the number of these replication events to just a few times per day. As a result, the RPO becomes measured in hours rather than the desired moments, rendering this approach, by itself, an ineffective strategy for low RPO.

Many businesses use virtual machine or disk replication, thinking that it will provide a stable recovery point. Many use these tools without pausing the database. It will work some of the time but, keeping our engineering credo in mind, our business expects us to create a recovery process that cannot be made to fail.

MYTH #4: WITH REPLICATION, WHO NEEDS BACKUPS?

In some cases businesses believe that since they were using hardware replication, there was no need for backups, trusting that a copy of the system would always be available on the surviving system. It then of course comes as a shock to learn that any undesired change of data state that occurred on the primary systems would quickly manifest itself on the backup system before they could prevent it. In addition, if the replicator or network failed, especially block-level replicators, the backup system could be left in a broken state. While it's never advisable to utilize a replication system in place of a backup system, it's an uncommon but real scenario that serves as a reminder that many businesses settle into a false sense of security based on a fundamental misunderstanding about the capabilities of a solution. As a business owner, it falls to you to demand proof that your are systems are recoverable.

SUMMARY

Cloning, snapshots, and disk replication can be useful tools. However, many organizations have deployed these technologies without recognizing their limitations. They may appear to work in simple tests, but the real measure of a recovery system is its ability to restore your processing capacity when your world is melting down. You must be sure that your backup and recovery methods cannot be made to fail in the most intensive use cases.

How Databases Become Corrupt

While we have reviewed and debunked common misconceptions about backups and hardware-level replication strategies, we will now discuss how Database Journaling and Replication solve many of these problems.

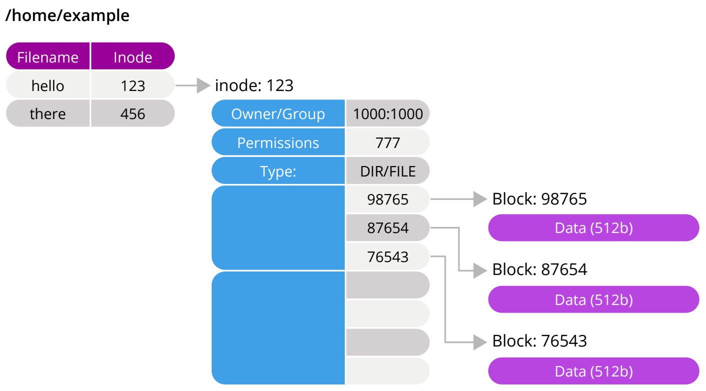
Terminology: MultiValue databases use the term "File" to refer to a database table. For clarity, we will use the term File/Table when referring to database storage and UNIX File when referring to an operating system file object.

How a database works

A database manages business level records and stores them safely onto the disk. It provides much better performance and stability than normal operating system disk files (UNIX files) and is specifically tailored to perform in scenarios of rapid data state change. Some databases store data within operating system UNIX file objects and others manage their own disk resources to create storage buckets. For the purposes of this paper, we will reference the operating system UNIX file method of storage. However, the same basic concepts with apply for any method.

How an operating system file works

Many data storage engines use operating system files as a storage object. Internally, a UNIX file looks like this:

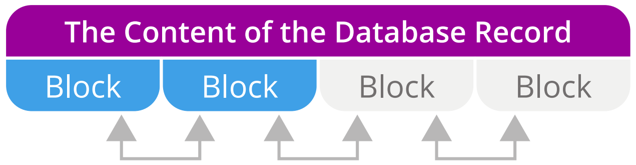


When a UNIX file is accessed by name, the file system looks up the name and returns an inode number which is how the file is identified within the system. The inode contains basic information about the UNIX file or UNIX directory including ownership, permissions, type, and a list of block numbers (disk block numbers) where the UNIX file's data is stored.

The blocks are normally 512 bytes in size. It may take thousands of 512-byte blocks to store the information in your database file/table. A database that uses UNIX files as storage objects will organize its data across these blocks. A database record may exist in one or many blocks. The database itself manages the blocks within its file/table.

When your application writes or reads data, it is concerned only with a record. It leaves it to the database manager to decide how to crack open the UNIX file that holds the database file/table's data. The application itself is not concerned about the concept of blocks.

Consider for a moment what will happen when a new, large record is written to a database file/table. The database will access the operating system UNIX file and then proceed to update blocks within that structure. Large records will require multiple blocks, so when that occurs, the database management system will link together several blocks, enough to hold the record. Each block may contain a link to the next block and a link to the prior block.



Because hardware level replication schemes cannot be aware of the relationship (linkages) between blocks, it is possible that, at precisely the moment in time that a replication event was triggered, not all the blocks in the database file/table may have been updated. The result is a 'broken' database file/table, because some of the blocks within its UNIX file had been updated while others had not.

The right two uncolored blocks represent blocks that needed to be updated to ensure integrity of the database file/table. Because the replicator is unaware of the relationship (linkage) between the blocks, it is possible that the snapshot/replicator could leave a file/table's blocks only partially updated resulting in a corrupt file/table.

By suspending updates from the database, flushing blocks to disk, and preventing further updates to occur (a database pause), this ensures that all unwritten blocks are committed to disk, ensuring the physical integrity of the database table/file's structure. However, most businesses cannot tolerate enough database pause/snapshot events during its workday to make this a practical strategy.

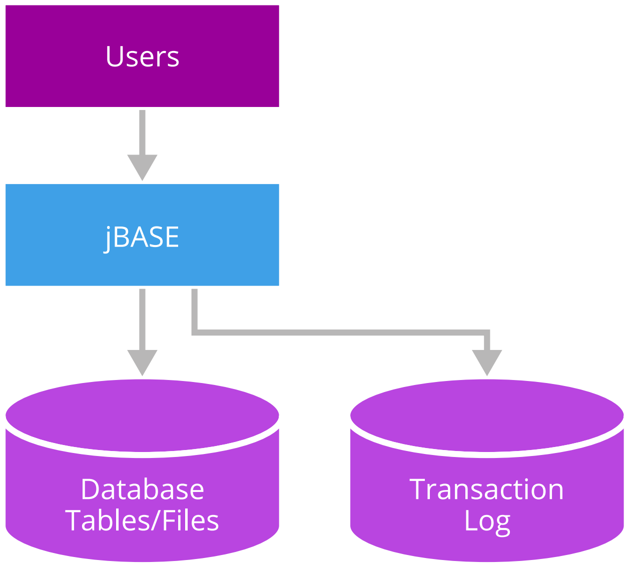
Transaction journaling

Most commercial disk or virtual machine replication strategies cannot ensure the integrity of database file/tables unless they take extra measures to ensure that the database file/tables are properly flushed and written before the replication can occur. We have also shown that due to the inconvenience caused to users and processes, continually pausing the database to ensure replication integrity is simply impractical.

Most database management systems (SQL, MultiValue, NoSQL, etc.) offer some form of transaction journaling. Most of them also allow you to use the transaction journals to achieve high-performance, resilient replication between servers. This technique is generically called log shipping, and it is the accepted practice for replicating databases.

How transaction journaling works

In its basic form, transaction journals record every update to a database, in chronological order, into a special log file. This log file documents the updates performed to the database in a format that can be re-applied to the database in the event of data loss. In this diagram, we'll use Rocket® jBASE to illustrate.



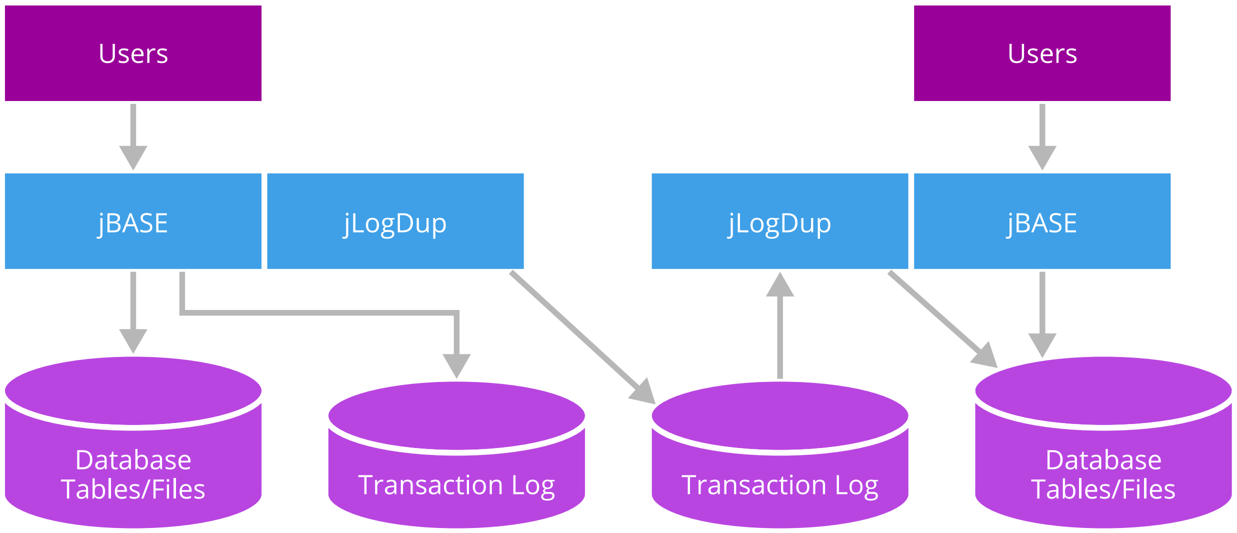
Running transaction journaling on a single server without replicating transaction journals to another system still yields substantial benefits in terms of RPO:

- If data is accidently deleted or modified, or the database is damaged, you can restore from your backup then roll-forward transaction journals to the time of the accident and recover with minimal data loss.
- If you need to see how the database looked at a specific point in time, you can restore from your backup and then play the transactions to that point in time and observe the data state. You can then selectively recover data from that image and move it to the main database.

← Back to contents

Using transaction logs to replicate to another server

This same log can be transmitted over the network to another Rocket jBASE server that can perform the same updates to its database in the same order. This second form of transaction journaling is called replication or log shipping and is equivalent to replication mechanisms used by other commercial databases.



Things to note about transaction logging and replication

Unlike hardware-level replication, Rocket jBASE transaction logging does not attempt to replicate disk blocks. Instead, it records the actual record update from the primary system and transmits it to the secondary/backup system. This approach provides the following benefits over hardware-level replication:

- Replicates entire records rather than blocks, thus eliminating any possibility of internal corruption due to partial block updates.
- Requires no database pauses or business interruptions to ensure the integrity of replicated records.
- Preserves the record update sequence to the database, ensuring that updates are applied to the backup database in precisely the same order as they were applied on the primary database.
- Uses transaction logs to recover the primary database and/or the secondary database.
- Replicates transaction logs to one or more replication targets, thus moving your recovery data to other places in case the primary server becomes inaccessible.
- Pauses transaction log recordings without affecting the primary or the secondary server.
- Reduces the volume of data transmitted between systems, allowing less latency (lag) between primary and secondary server updates.
- Recovers a database by restoring a backup then applying transaction logs to the database to bring it back “up to date” thus achieving very low RPO.
- Provides record-level replication (rather than block-level), assuring physical integrity of files on backup server without the need for business interruptions required during a database pause.
- Enables RPO objectives of moments, whereas other approaches can only achieve assured RPO measured in hours.

← Back to contents

How to know if your business would benefit from database replication?

- Does your business require or desire to continue business as normal in the event of a server failure?
- Does your business require or desire little or no data loss resulting from a server failure?
- Would the impact on your business be great enough that it would be worthwhile to invest some amount of money into mechanisms and procedures, to reduce or eliminate the possibility of system unavailability or data loss?

For most modern, connected businesses, the answer to each of the above questions is yes. The cost to implement these protection measures has never been lower, nor the process simpler. Rocket jBASE's replication and recovery strategies are implemented around the world, in thousands of banks that maintain the highest expectations for RPO and RTO. We recommend that any Rocket jBASE user give Transaction Journaling and Replication serious consideration.

Conclusion

There are two types of small/medium business organizations:

- Those that understand that low RPO and RTO are not luxuries but necessities in today's business climate, and have implemented proper backup, replication, and recovery procedures.
- Those that do not understand the value and are destined to encounter debilitating failures and data loss due to lack of proper planning.

For those in the latter camp, it is only a matter of time before a failure causes significant organizational damage and hardship. Even if an organization is not prepared to implement a backup server with replication for low RTO, it must implement Transaction Journaling, at least on its primary business server, to achieve a reasonable RPO. Data loss is no longer an acceptable risk. By journaling transactions properly, your organization can be protected.

Next Steps

- If you are unsure of your backup and recovery strategy, please reach out to your Rocket Software representative for a free initial consultation.
- If you are unsure of your current RPO or RTO capabilities and would like to explore this with an expert, please reach out to us.
- If you have procedures that have not been reviewed and verified in a while, now is the time. If you want a third-party to help you independently validate your backup and recovery strategies, we are here to help.

About Rocket Software

Rocket Software partners with the largest Fortune 1000 organizations to solve their most complex IT challenges across Applications, Data and Infrastructure. Rocket Software brings customers from where they are in their modernization journey to where they want to be by architecting innovative solutions that deliver next-generation experiences. Over 10 million global IT and business professionals trust Rocket Software to deliver solutions that improve responsiveness to change and optimize workloads. Rocket Software enables organizations to modernize in place with a hybrid cloud strategy to protect investment, decrease risk and reduce time to value. Rocket Software is a privately held U.S. corporation headquartered in the Boston area with centers of excellence strategically located throughout North America, Europe, Asia and Australia. Rocket Software is a portfolio company of Bain Capital Private Equity. Follow Rocket Software on LinkedIn and X (formerly Twitter).

Modernization. Without Disruption.™

Visit RocketSoftware.com ›

Learn more

Rocket software

MAR-9544_WP_jBase_ProtectingYourBusiness_V4