

High Availability—Always On Technologies

White Paper

Published: November 2007

Updated: July 2008

**Summary:** Microsoft SQL Server 2008 reduces downtime of business-critical databases by providing a wide range of high availability solutions. These solutions result in data being available to customers and employees twenty-four hours a day, seven days a week, and still deliver cost-effective high performance. The Always On features in SQL Server 2008 increase protection against system and hardware errors, reduce the number and recovery times of planned or unplanned downtime, and decrease conflicts caused by large numbers of users who are accessing the same database.

For the latest information, see [Microsoft SQL Server 2008](http://www.microsoft.com/sqlserver/2008/en/us/default.aspx).

Contents

Introduction 1

Availability Scenarios 1

Increased Availability 2

Database Mirroring 2

Log Shipping 2

Failover Clustering 3

Geographically Dispersed Failover Clustering 4

Peer-to-Peer Replication 4

Decreased Downtime 4

Fast Database Recovery 5

Backup and Restore 5

Checksum on Data Pages 6

Online Index Operations 6

Online, Piecemeal, and Page-Level Restore 6

Partial Database Availability 6

Snapshot Isolation 7

Dynamic Configuration 7

Enhanced Manageability 7

Database Snapshots 7

Table and Index Partitioning 8

Backup and Restore 8

Dedicated Administrator Connection 8

Resource Governor 8

Conclusion 9

# Introduction

Application downtime can adversely affect business and result in lost revenue, customer dissatisfaction, and damage to credibility. Many database applications, especially enterprise business applications, require minimum system downtime. System downtime can be caused by a variety of issues such as human error, natural disaster, hardware or software failure, and application upgrade. Microsoft® SQL Server® 2008 Always On Technologies provide a full range of options to minimize downtime and maintain appropriate levels of application availability.

## Availability Scenarios

Achieving high availability is a common requirement in many businesses. The appropriate solution for a particular scenario depends on a number of factors, and the solution that works best in one case might not be the most suitable solution for another case. For example, consider the following scenarios.

* An Internet-based sporting goods retailer wants to provide near-continuous availability to its customers. The Web servers are in a Web farm, but database failures cause errors and most of these situations result in the company losing customers. The company has an approved standard server build and ideally wants to avoid major costs, infrastructure changes, or application re-writes.
* A recruitment company is concerned about user error on its client database. The company would also like to have a warm standby server that could be brought online quickly although instant failover is unnecessary.
* An insurance company wants to provide near-continuous availability for its claims database. The performance of the system must not be affected and documentation to support the claim, which is currently stored in the file system, must also be made available.
* A management training company has three sites with a sales team at each site. Each sales team enters records into the Orders database for its own site. To improve performance, each site requires access to all orders for the company, ideally on its local server. In the event of a failure, to provide availability, each site must be able to switch to the server at another site.
* An accountancy company wants to provide availability to its clients, even if there is a wide-ranging failure such as a natural disaster. Information does not need to be made instantly available, but it should be available within hours rather than days. The company would also like to retain a daily version of its databases for regulatory compliance. The copy data does not need to be instantly available, but must be available for a number of years.
* A toy manufacturer wants to run reports against its sales database but, because this is constantly in use, reports are often blocked by currently running transactions. The company has looked into reporting against uncommitted transactions, but this has resulted in incorrect results because some of the transactions did not complete.
* A real estate company wants to protect its system against user error by being able to roll records back to the values of the previous day. The company would also like to be able to run reports against yesterday’s data that are unaffected by transactions taking place today. Ideally, the company would like to implement a solution that has the minimum possible disk overhead.

With such a wide range of availability requirements, businesses need a database platform that provides flexible options for high availability solutions. This white paper shows you how SQL Server 2008 Always On Technologies provide these flexible options for high availability, and enables businesses to implement the best solution for their particular requirements.

# Increased Availability

Many database applications are mission-critical, and therefore demand minimum downtime and quick recovery in the event of a disaster. SQL Server 2008 offers a full range of Always On Technologies to minimize downtime and achieve the appropriate level of availability.

## Database Mirroring

*Database mirroring* is a technology for increasing database availability. As changes are applied to the principal database they are automatically applied to the mirror. In the event of a failure on the principal server, client applications are automatically redirected to the mirror server without any changes to the application.

You can configure database mirroring to apply changes to the mirror in real time, known as *synchronous*, or near real time, known as *asynchronous*, so you can specify the level of protection and the performance overhead on your system.

With SQL Server 2008 Enterprise, mirroring protection is extended to pages of data. If a page is found to be corrupt on either the principal server or the mirror server, the corresponding page is retrieved from its partner server and database operation continues seamlessly. SQL Server 2008 also improves database mirroring by compressing the data flow between principal and mirror servers and by removing the need for a database restart if a manual failover is performed.

In the scenario of the Internet-based sporting goods company, database mirroring would provide a very fast failover in the event of a server, database, or even page-level failure. Applications would not need to be rewritten and, although a new server would have to be purchased, the standby server would be standard equipment and would not require a SQL Server license.

## Log Shipping

*Log shipping* is a high-availability technology that provides a warm standby server. A back up is taken of the master server and restored on one or more secondary servers. Log shipping then applies periodic log backups to the secondary servers. While database mirroring can only have one mirror server, log shipping can have many secondary servers, which increases the level of protection. Log shipping can also take advantage of backup compression to reduce the size of the log files.

Log shipping occurs on a schedule and therefore there is a time delay between data changing on the master server and these changes transferring to the secondary server. This delay can result in data loss, but in some scenarios the delay can be used as a means to undo user errors by delaying the application of log data to the secondary server (leaving the original data on the secondary server).

To reduce the workload of the primary server, the secondary server can act as a read-only reporting server, although reports cannot be run while log backups are being applied.

In the scenario of the recruitment company, log shipping could be applied with a delay between the log backups and the restore on the standby server. This allows user error to be reversed and, if there is a failure on the primary server, the logs could be applied straightaway to reduce data loss.

## Failover Clustering

*Failover clustering* is a Microsoft Windows® feature that provides protection to the entire server, not just the databases. Multiple servers, known as *nodes*, share a disk array so if one node fails, its services can be started on another node in the cluster. This solution prevents any data loss and provides automatic client redirection, although the failover from primary to secondary server is not as fast as database mirroring and specialized hardware is required.

Microsoft Windows Server® 2008 extends the clustering abilities of previous versions of Windows both by reducing hardware and infrastructure requirements and by supporting 16 nodes in a cluster. Overall, clustering is easier to set up and manage in Windows Server 2008, and the introduction of the cluster validation tool helps to ensure that you have adequate hardware resources for a clustered solution. By taking advantage of the clustering enhancements in Windows Server 2008, SQL Server 2008 provides a robust high availability solution for an entire SQL Server instance.

In the scenario of the insurance company, Failover clustering could be used to provide near continuous availability. Failover clustering does not affect system performance and, because it is a server level solution, it can provide availability to other resources on the server. Geographically dispersed failover clustering could be used if there was a concern that the shared disk array did not provide sufficient redundancy.

## Geographically Dispersed Failover Clustering

*Geographically dispersed failover clustering* provides server-level redundancy on a certified Microsoft Geographically Dispersed Cluster Services configuration with SAN replication and a VLAN. If the site, the server node, or the disks fail, the complete redundancy of systems and disks enables the failover cluster to handle subsequent activities on another site. This configuration removes the risk of failure of a shared disk array, which would prevent a standard cluster configuration from working. Server nodes do need to be on the same the subnet and SQL Server 2008 does not support OR dependencies between IP addresses.

## Peer-to-Peer Replication

*Peer-to-peer replication* enables multiple databases to be replicated with each other. Changes are allowed on any database and can be applied to other nodes in the replication topology in close to real time. Applications must be designed to connect to another node in the topology if their principal server is unavailable.

SQL Server 2008 introduces enhancements to peer-to-peer replication by enabling the addition of nodes to a replication solution while keeping the replication process online. In previous releases of SQL Server, the replication process had to be taken offline to add new nodes. Additionally, SQL Server 2008 makes the set up, monitoring, and management of peer-to-peer replication straightforward with a graphical Topology Viewer. SQL Server 2008 also introduces conflict detection, so you can protect against accidental conflicts when multiple replication nodes update the same row.

In the scenario of the management training company, peer-to-peer replication could provide local access to the entire orders database and each of the sites could be used as a failover if the database of another site becomes unavailable.

# Decreased Downtime

System downtime is a threat to the success of any business. A system downtime may be unexpected or may be planned for maintenance tasks, such as rebuilding an index or performing an application or system upgrade.

Protection of your systems against unexpected downtimes is much more challenging because any unexpected hardware or network failures, human error, natural disaster, or theft could be catastrophic to the business. SQL Server Always On Technologies enable you to recover quickly from an unplanned downtime. In addition, with the proper implementation of Always On Technologies, system maintenance and operational procedures that are necessary for smooth operation of mission-critical applications can occur with minimum downtime.

## Fast Database Recovery

The SQL Server 2008 Enterprise Database Engine provides fast recovery during both crash recovery and database mirroring failover. Fast recovery makes the database available during the undo phase of the recovery process, with partial availability during restore operations, database page checksums, and backup media mirroring. By supporting fast recovery, SQL Server 2008 minimizes the time taken to bring a database online, so the database is available to users sooner than if the entire recovery process had to be completed before users and applications can connect.

## Backup and Restore

Backups are essential in a high availability solution to provide periodic snapshots of the data and as a source to restore the data in case of wide-ranging failures or a lack of other high availability solutions.

The backup media mirroring feature of SQL Server enables you to perform a mirrored backup of a database to multiple backup devices, which greatly increases the reliability of backups in case of faulty media or a lost backup device. SQL Server 2008 supports mirrored backups to both disk and tape, but the devices used must be similar to avoid the risk of a device mismatch error. All devices in a backup mirror set must be available during the backup process, but the database can be restored from any individual backup in the mirror set. To perform a mirrored backup with Transact-SQL, you must use the MIRROR TO clause of the BACKUP command as shown in the following example.

BACKUP DATABASE AdventureWorks

TO TAPE = '\\.\tape0', TAPE = '\\.\tape1'

MIRROR TO TAPE = '\\.\tape2', TAPE = '\\.\tape3'

WITH

FORMAT,

MEDIANAME = 'AdventureWorksSet1';

GO

Another backup reliability feature of SQL Server is the generation of a checksum during backup operations, which is then used to verify a restored database.

SQL Server 2008 includes support for backup compression, which can reduce the volume of backup sets by up to fifty percent. The reduction in volume enables you to backup data more frequently which, in turn, reduces restore times. Backup compression can be set to ON or OFF in a BACKUP command, and you can use **sp\_configure** to define the default compression value (*on* or *off*) for when a BACKUP is taken without specifying the compression option.

## Checksum on Data Pages

*Checksums* compare the values that were written to disk with the values that are subsequently read. If these values do not match, the page is marked as suspect and either restored manually or, with SQL Server 2008, automatically restored from a partner server.

## Online Index Operations

Availability can be affected by standard maintenance operation as well as failures. Index maintenance has traditionally caused the affected data to be unavailable, but, from SQL Server 2005 onwards, data remains available during Online Index Operations such as creating, dropping, or rebuilding an index.

Online Index Operations increase availability of all of the systems when index maintenance is performed, so increases staff productivity and improves the customers’ experience of your systems.

## Online, Piecemeal, and Page-Level Restore

SQL Server 2008 Enterprise supports online restore, which enables you to access restored portions of a database even if the remainder of the database is not yet available, which speeds up recovery, reduces downtime, and increases availability. Additionally, you can use a *piecemeal restore* strategy to restore a database in stages; for example, you can restore file groups that contain currently active data and bring the database online before you restore file groups that contain older, archive data. The piecemeal restore strategy reduces the amount of time mission-critical applications remain offline in a disaster recovery scenario.

When you need to recover a database because of a small number of corrupt pages, SQL Server 2008 supports page-level restore operations. You can identify the page ID of suspect pages in the database from a number of sources, including the **suspect\_pages** table in the **msdb** database, the SQL Server error log, and the DBCC CHECKDB command. Having identified any pages that are potentially corrupt, you can restore them by specifying the page ID in the PAGE clause of the RESTORE statement. The ability to restore individual pages from a backup can significantly reduce the downtime caused by database corruption.

## Partial Database Availability

*Partial database availability* enables a database to remain online and available even if part of the database is damaged due to isolated hardware or disk failures. If secondary non-primary data files are damaged, the undamaged portion of the database remains online and available. So data is available to some customers and staff even if the whole system is not available, which reduces the effects of a problem while another availability solution comes online.

## Snapshot Isolation

Traditionally during data modifications, data that is being modified is locked so that it is unavailable to other applications and users. The alternative approach is to read the current state of the data, even though those values are uncommitted and may be rolled back if the transaction does not complete successfully.

*Snapshot isolation* extends the locking framework in SQL Server 2008 by enabling applications to view the value before any data modifications took place. This prevents the application from being blocked, while still providing genuine committed data. For example, in the scenario of the toy manufacturer described at the beginning of this white paper, snapshot isolation would enable the reports to run, while providing transactionally accurate results. Snapshot isolation results in more timely and more accurate reporting.

## Dynamic Configuration

*Dynamic configuration* is one of the SQL Server Always On Technologies that offer better access to hardware features, such as hot upgrades, while the system is running. When your hardware supports hot-add memory, system memory can be increased without interruption and SQL Server automatically uses this memory through dynamic memory and even dynamic Address Windowing Extensions (AWE) for failover clusters.

SQL Server 2008 also supports hot-add CPU, so you can add processors to supported hardware with no interruption in activity.

For any system that requires processor or memory resources to be increased, dynamic configuration reduces downtime and increases both staff productivity and customer satisfaction.

# Enhanced Manageability

SQL Server 2008 delivers powerful, productive management tools that simplify recovery by providing tools to investigate, monitor, and recover damaged data across your enterprise.

## Database Snapshots

Whether caused by maintenance operations or end user interaction, human error is the most common cause of downtime. SQL Server Always On Technologies provide *database snapshots*, which enable quick investigation and recovery from user errors. Database snapshots are read-only, static views of a database at the moment of snapshot creation. Database snapshots can also be used for reporting without affecting the availability of the source database.

In the scenario of the real estate company, database snapshots provide a method of rolling back current values to the day before and also provide a read only reporting solution for yesterday’s data. Because database snapshots only store data for the changes that are applied to the live database, they are very cost effective in terms of storage.

## Table and Index Partitioning

*Table and index partitioning* improve availability and manageability of large tables by breaking them into smaller, more manageable units called *partitions*. Partitions can be accessed independently of each other, so limit the effect of I/O-intensive activities such as data load, backup and restore, index rebuild, and index defragmentation.

SQL Server 2008 increases concurrency on partitioned tables by enabling row and page locks to be escalated to partition locks. This escalation results in less locks being applied and therefore speeds up spanning queries. In previous versions, row and page locks were only escalated to table locks, which could lead to an unnecessarily large lock being applied and that could block other users.

For any system that includes large tables and large numbers of users, increased concurrency and reduced contention improves availability and results in improved staff productivity and customer satisfaction.

## Backup and Restore

Backup and restore operations provide more flexibility in recovery. SQL Server enables you to restore a read-only database file group without applying transaction logs. It also allows a database backup and a transaction log backup to run simultaneously.

## Dedicated Administrator Connection

The *Dedicated Administrator Connection (DAC)* is a high priority connection to the server that helps with troubleshooting when the server appears to stop and does not accept new connections.

## Resource Governor

In highly used databases, different workloads can compete for the available system resources such as CPU threads and memory. In some cases, this resource contention can cause mission-critical workloads to be adversely affected by other activity in the database. SQL Server 2008 includes *Resource Governor*, which you can use to proactively manage workload resource utilization by assigning priorities to different workloads based on logins, application, and other factors. The ability to prioritize workloads results in more predictable performance, and can help to ensure that database server resources are always available for mission-critical applications.

# Conclusion

Any system that is critical to your business immediately affects your profitability from the moment it becomes unavailable to customers and employees. There are a wide range of availability problems and requirements, and one solution cannot resolve all of these issues.

SQL Server 2008 provides a wide range of solutions to improve availability that are at least as good, in terms of performance and variety, as anything available from any of the major database vendors and SQL Server 2008 continues to lead the way in terms of manageability and value for money.

For more information:

Microsoft SQL Server 2008  
<http://www.microsoft.com/sqlserver/2008/en/us/default.aspx>

**Please give us your feedback:**

Did this paper help you? Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

* Are you giving it a high rating because it has good examples, excellent screenshots, clear writing, or another reason?
* Are you giving it a low rating because it has poor examples, fuzzy screenshots, unclear writing?

This feedback will help us improve the quality of white papers we release. [Send feedback](mailto:sqlfback@microsoft.com?subject=White%20Paper%20Feedback:%20%5bTitle%20of%20white%20paper%5d).