Complete Online Microsoft SQL Server Data Protection

VERITAS BACKUP EXEC™ 9.1 FOR WINDOWS SERVERS

Agent for Microsoft SQL Server
SQL Server 7.0
SQL Server 2000
# TABLE OF CONTENTS

Executive Summary ................................................................................................................................................... 3  
Usability ............................................................................................................................................................. 3  
Reliability ........................................................................................................................................................... 3  
Why Protect Microsoft SQL Server? .......................................................................................................................... 4  
Why Do you Need the Backup Exec Agent for Microsoft SQL Server? ................................................................. 5  
Protecting SQL Server ............................................................................................................................................... 6  
Introduction ............................................................................................................................................................ 6  
Applicaton Protection ................................................................................................................................................. 6  
  Business Requirements ........................................................................................................................................ 6  
  Options............................................................................................................................................................... 8  
  Protecting Clustered SQL Servers .................................................................................................................... 8  
  Deployment Guidelines...................................................................................................................................... 8  
Database Protection ................................................................................................................................................... 9  
  Determining your Database Protection Needs for SQL Server ......................................................................... 9  
  SQL Server's Storage Layout ................................................................................................................................ 9  
  Business Requirements ........................................................................................................................................ 11  
  Backup Methods and their Impact During Restore.......................................................................................... 11  
  Options............................................................................................................................................................. 13  
  Deployment Guidelines.................................................................................................................................... 15  
Additional Microsoft SQL Solutions from VERITAS Software .................................................................................. 17  
Summary .................................................................................................................................................................. 18
EXECUTIVE SUMMARY

VERITAS Backup Exec™ Agent for Microsoft SQL Server ensures your business critical Online Transaction Processing (OLTP), Online analytical Processing (OLAP), and e-business data is protected in case of application or hardware-based corruption or loss. Designed with flexibility and ease of use in mind, this option will give SQL Server 7 and SQL Server 2000 users complete and customizable protection down to the individual table space or filegroup. Is the backup window too small for a full backup? This option can perform differential backups as well as Transaction Log backups with automatic truncation. Restoring to another SQL Server machine is easy, because this option can redirect a restore.

The Agent for Microsoft SQL Server supports “rollback” and “single-pass” restores, allowing administrators to restore databases based on a point in time, rather than a specific backup job. Multipipe support along with SQL 7 Virtual Device Interface (VDI) gives users the easiest and fastest way available for complete SQL database protection.

PRODUCT HIGHLIGHTS

For SQL Server 2000
- Data recovery to named Transaction Log Marks within the transaction log allows administrators to restore data up to the point at which the data had last been committed to the database.
- Modeling of SQL database backups can be targeted to fit the individual needs of the business by performing full or differential backups and restores of the filegroup.
- Expanded data protection parameters include multiple and named SQL Server 2000 database instances running on the SQL Server database.
- Improved performance of database consistency checks (DBCC) with the ability to perform a physical only DBCC on SQL 2000 databases greatly enhances backup speeds without sacrificing backup accuracy.

Usability
- Transiently integrates online or “hot” SQL Server backups within regularly scheduled network protection routines.
- Individual table space or individual filegroup backup and restore.
- Microsoft SQL Server supports “rollback restores” to a specific point in time, rather than a specific backup job.
- With flexible backup launch options for SQL Server, backup jobs can be launched immediately or scheduled.

Reliability
- Uses the native SQL Server APIs for both backups and restores for assuring reliable and consistent SQL Server protection.
- Integrates with the VERITAS Backup Exec Intelligent Disaster Recovery™ Option for a rapid, bare-metal system disaster recovery to the last full, incremental, or differential backup, complete with identical configuration of the operating system, user profiles, updates, and other applications.

Key Benefits
- Supports both 32 bit and 64 bit SQL Server installations
- Helps maintain the integrity of vital SQL Server data.
- Incorporates online nondisruptive SQL Server database protection as part of everyday backup routines, which increases the chance of data recovery and minimizes data loss without inhibiting daily database activity.
- “Rollback,” “Single-Pass,” and Transaction Log Mark restores allow administrators to recover the database to a point in time or last commit point, allowing for quick and reliable data restores.
WHY PROTECT MICROSOFT SQL SERVER?
Microsoft SQL Server™ is a general purpose relational database server which can scale from hosting simple databases to supporting clustered, mission critical business applications such as SAP. In fact, SQL Server is the most popular relational database on Microsoft Windows, with 38 percent market share. (Gartner, June 2001) Simply put, the more your business depends on SQL Server, the more important it is to protect it.

In order to maintain Microsoft SQL Server's availability and protect its databases, a working and thoroughly tested data protection and recovery plan and reliable data protection software are essential. Together, they can ensure the recovery of the SQL Server environment and its databases. The key objective is to help minimize downtime for your database environment and to provide the quickest possible data recovery in the event of a system crash, database corruption, or other forms of data loss.

This white paper will address several aspects of a SQL Server data protection plan, focus in on how VERITAS Backup Exec 9.1 for Windows Servers and the Backup Exec Agent for Microsoft SQL Server can meet the needs of this plan, and introduce several other VERITAS products which enhance SQL Server data protection and availability.
WHY DO YOU NEED THE BACKUP EXEC AGENT FOR MICROSOFT SQL SERVER?

Protecting a database server like Microsoft SQL Server requires careful thought and planning in order to meet the availability needs of your company and its budget. The most common method of formalizing these needs is Service Level Agreements. Basically, these agreements are contracts between the users and providers (e.g., IT departments), which outline such factors as: expected services, acceptable downtime, and response time for problem resolution. It is critical that you understand these factors during the design phase of your SQL Server deployment, as they can heavily influence the resources you’ll need to support the plan.

The basic rule of thumb regarding data protection is the higher the requirement for availability, the higher the cost will be. The chart above illustrates this concept and shows the various technology stops along the way toward higher availability. You will notice that the cornerstone of any availability solution is backup, and choosing a reliable backup product should be paramount since it may be your last line of defense against data loss. VERITAS Backup Exec together with the Agent for Microsoft SQL Server easily meets the criteria for fast, flexible, and reliable SQL Server data protection. In fact, Backup Exec has supported Microsoft SQL since its introduction to Windows NT in 1995 (and has supported Windows NT and Windows 2000 since their introduction)...so VERITAS has significant experience in this market.

In addition to offering two products (Backup Exec and NetBackup) which support the basic level of availability for SQL Server, VERITAS also develops several other products which support your SQL deployment on up through the highest levels of availability. This White Paper will briefly introduce these products.
PROTECTING SQL SERVER

INTRODUCTION
With most database applications like SQL Server, data protection can be divided into two main objectives – preparing for a disaster recovery where all data (Windows operating system, SQL Server application, and its databases) are destroyed, and preparing for the restoration of all or some of the user database data.

Disaster recovery preparation is comprehensive and includes protecting SQL Application (Windows operating system, System State, the SQL Server application directory, and SQL Server’s system databases) and SQL Server’s User databases. While this white paper will briefly outline the data disaster recovery preparation steps to protect SQL Server’s application files, you should read the following Backup Exec white paper which focuses on protecting the Windows Operating System: “Enhanced Data Protection for Windows Servers”. These white papers can be located at http://www.veritas.com.

SQL Server provides several ways to deploy and organize User databases and logs, along with several methods to backup and restore them. Each of these choices can affect the granularity and speed at which you can restore your data, so it is necessary to understand the pros and cons between them in order to achieve your data protection goals. This document will cover these topics and help you understand which data protection scheme is right for you.

APPLICATION PROTECTION
At the application protection level, the focus is to protect the SQL Server’s application files and configuration, which includes SQL Server’s system databases. The goal is to prepare yourself for such cases when you need to simply restore some SQL Server settings that were erroneously made, or to ensure you are prepared for a successful disaster recovery of SQL Server. Listed below are a few requirements, options, and guidelines to protecting SQL Server.

Business Requirements
Backup the host server for SQL Server – Since SQL Server runs on Windows 2000, protecting the underlying Windows operating system and SQL Server’s files and settings are very important for a timely disaster recovery. This includes backing up all files on the volumes that Windows and SQL Server are installed on, and backing up the Windows System State (which includes the Windows registry). The backup schedules of this data should be coordinated with the backups of SQL Server User data (outlined below) so that you have a consistent set of data for an easier disaster recovery.

Backup Exec Advantage
With Backup Exec, you can easily protect Windows files, Windows System State, SQL Server files, SQL Databases (including the system databases) within a single schedulable job; or you can break these tasks up into multiple jobs if your environment, performance needs, schedule, or data retention periods demand. If disaster occurs to your SQL server, the Backup Exec Intelligent Disaster Recovery (IDR) option can help you quickly bring Windows back online in preparation of performing a DR of SQL Server.
Backup Exec 9.1 clearly displays all SQL Server data and allows you to easily integrate database backups into your backup scheme.

Backup SQL Server's System Databases – SQL Server itself uses 4 system databases for configuration and operation.

- **Master Database** – this is the most important of the system databases – it must be backed up. The Master database is like the registry within Windows 2000 to SQL Server - it contains configuration information on SQL Server parameters, user databases, security, stored procedures, and a multitude of other critical data that SQL Server relies on. While the Master database can be backed up like any other SQL database, the restore process is not trivial - it requires that SQL Server be put in a special mode (single user) before restoration so that no other users are accessing SQL Server, then returned to normal mode afterwards. Thus the recovery process requires special procedures before and after the restore process to properly recover the Master database.

- **MSDB, Model, and Distribution Databases** – these system databases are less critical than the Master Database, but should still be routinely backed up. Briefly, the MSDB database is the scheduling database for SQL Server's internal operations; the Model database is the template from which all new user databases are based. The Distribution database contains information about the replication operations (Distribution only exists if you setup SQL Replication). Like the Master database, these system databases can be backed up like any other database, but have constraints during restoration, but they do not require SQL Server be placed in single user mode.
Backup Exec Advantage

In addition to fully supporting the backup and restore of SQL Server’s system databases in an easy to use interface, Backup Exec 9.1 includes two features that further automate the protection of system databases.

- **Automated Master Database Restore** – this industry first feature reduces the complexity of a Master database restore by automating the manual steps you normally would have to do. It even supports SQL Server in a Clustered configuration.

- **SQL Server Intelligent Disaster Recovery Integration** – Normally, recovering SQL Server after a complete Disaster Recovery requires two steps: Recover SQL Server’s System Databases and then recover the User databases. With Backup Exec, this is reduced to one step: Recovering the User databases. Backup Exec backs up the system databases during file backups as off-line files (system databases are usually small) and the Intelligent Disaster Recovery Option restores them – allowing you to skip a step in the disaster recovery process and save valuable time.

Options

Protecting Clustered SQL Servers

An enterprise level feature of SQL Server is its tight integration with Microsoft Cluster Services (MSCS). MSCS offers the huge benefit of clustering two or more Windows 2000 or Windows Server 2003 servers (called nodes) to serve as one highly available server in the event one server becomes unavailable. With MSCS, SQL Server presents itself as one “Virtual Server” which can actually represent all of the servers in the cluster. To properly protect a clustered SQL installation, the backup application must be able to target the Virtual Server, so that if one SQL server fails, the backup and restore operations can continue.

Backup Exec Advantage

Backup Exec fully supports up to an eight-node cluster of SQL on Windows 2000 and Windows Server 2003 (eight is currently the maximum number of nodes MSCS offers). Backup Exec can automatically restart database backups that were interrupted because of a failover.

Deployment Guidelines

- **SQL Server system database backup**: The Master database should be backed up before and after any significant changes to the SQL Server configuration (e.g., adding/deleting databases, users, stored procedures, or changing database storage). Since SQL Server’s system databases (Master, MSDB, Model, and Distribution) are usually small, including them in a routine daily backup can save you much time and headache if a restore is needed.

- **Disaster Recovery Tip**: In order to restore a consistent snapshot of backup data during disaster recovery, a good strategy is to coordinate the full backups of the Windows operating system files, SQL Server application files, and the Windows System State with the full backups of SQL Server’s databases. Follow this strategy for differential or incremental backups of files with differential and log backups of SQL databases too.
DATABASE PROTECTION

Determining your Database Protection Needs for SQL Server

SQL Server is a highly scalable relational database platform that can host a single database that is only a few megabytes, to a multi-terabyte set of interdependent databases in which a business critical application like SAP relies. To meet this scalability challenge, SQL Server offers several ways to deploy and protect your databases depending on your business and availability needs. To understand these needs, you should have answers to the questions below so you can create a SQL Server installation that will meet your needs today, and provide expandability as your needs grow.

Questions to help you estimate your general availability and data protection requirements:

- Is the data that SQL Server will host under an existing Service Level Agreement (SAL)? If so, what are the data protection requirements?
- What are the availability requirements? What are the tolerable limits that the database can be offline each day?
- If you experience disk or network failure, what is the acceptable downtime?
- What is the acceptable downtime in case of a complete disaster? Will you need to replicate the database and provide clustering to failover to another site?
- In the event of a disaster, which databases should be available first? Who should manage the storage for SQL Server, backup administrators or SQL administrators?

Questions to help you estimate specific data protection requirements:

- What is the size of each database today? What is the growth rate of the database?
- How often does the data in each database change? Do you have tables which static data?
- Which hours during the day do your users demand the best performance from SQL Server? Are the other hours available for backup?
- Do you have enough space for transaction log growth during heavy database activity?
- Do your tape drives have enough capacity and performance to backup or restore your largest databases in the allotted time window? Will you need to consider backup to disk for staging or better performance?

SQL Server's Storage Layout

Once you understand the SQL Server availability and data protection requirements, then you are ready to consider the storage layout of your SQL Server databases. Following is a brief description of the major parts of SQL Server storage and their usage:

Database: A collection of information, tables, and other objects. Databases can be contained in one single file, or they can be split up to contain subsets of database data. Databases can be setup to automatically expand when needed.

Transaction Log: A file containing a running grouped list of ALL database transactions. There is one log file for each database. SQL Server uses these logs to recover from database errors and can be either wholly committed or rolled back (erased) to/from the database. You could think of transaction logs as an incremental backup of a database, since log files contain all changes to a database. Each transaction group has a time stamp and can also be named, allowing a highly granular restore to a particular point in time. Transaction logs must be periodically managed (truncated) to ensure they do not consume all available log disk space. Although databases can be configured to not maintain a log (i.e. SQL’s System Databases), it is not recommended due to the limitation of only restoring to the last Full or Differential backup.
**Filegroup:** A group of database files. By default, a database belongs to a Primary Filegroup, but SQL Server allows you to split up a database into multiple files. These files can be organized into multiple Secondary Filegroups that provide the following advantages:

- **Increase Storage Flexibility:** Filegroups allow you to place specific data (tables) on volumes that can be easily expanded when storage needs arise. This allows you to break up a large database into smaller files that can be managed easier. SQL Server fills all database files in a filegroup evenly, so eventually all will be the same size.

- **Increase Performance:** Filegroups allow you to split a database’s files across multiple physical drives, which can increase performance of the database. This also has the effect of balancing the load (I/O bandwidth) across multiple drives. In addition, if only one drive fails, you may only need to restore that drive’s data and any log data to recover vs. restoring the entire database.

- **Increase Availability by Isolating Database Activity:** Filegroups allow you to place static tables in their own database file, which can then be backed up on a less frequent schedule than the rest.
Business Requirements

Hot (on-line) Backup and Restore of the SQL Server User Databases
SQL Server provides several methods that data protection software vendors can use to backup and restore SQL Server databases while they are on-line. The methods depend on the availability and performance needs of the database along with the way you have chosen to configure the database. We will cover the backup and restore methods and then discuss when to use them.

Backup Methods and their Impact During Restore

Full Database Backup – Backs up the entire selected database. Full Backups usually transfer the largest amount of data and thus consume more time and resources than other backup methods. However, they are the foundation of which all of the other backup types are based from – they must be accomplished. Full backups are usually followed by differential and/or transaction log backups.
- **Restore Impact:** Restoring a full database backup will take longer than other backup methods, but once you are finished, the database is ready to be brought on-line.

Differential Database Backup – Backs up only the changed blocks (extents) within the database since the last full database backup. Since this method only backs up the changed blocks, it is very space efficient and provides a quick way to backup the differences in a large database from the full database backup. Differential backups are usually followed by transaction log backups.
- **Restore Impact:** The main advantage to using differential database backups is during restore: you only need to restore the full database backup and the last differential database backup (since they are cumulative) to fully recover the database. For example, if full backups are performed Sunday and differentials during the weekdays, then only 2 (1 full + 1 diff) sets of data would be needed to recover from a disaster on Friday. The disadvantage to using only Full and Differential backups is that you cannot recover to a specific point in time like you can with transaction log backups.

Transaction Log Backup – Backs up the Transaction Log for the selected database. You could think of transaction logs as an incremental backup of a database, since log files contain all changes to a database. Transaction log backups are typically larger than differential database backups, but are still a very efficient way to incrementally backup the database. There are two versions of Transaction Log backups: Truncate and “No Truncate”. The difference is that the Truncate version deletes or flushes the uncommitted transactions from the log after the log backup is successful, while the “No Truncate” does not. Normally, you will only use the Truncate version, as it is the best way to manage the transaction log size. The “No Truncate” version is typically only used when the database is corrupted or off-line with a problem and it provides a way to recover from disaster.
- **Restore Impact:** During a full database disaster recovery, you would normally restore a full database backup, a differential database backup (if accomplished), and then any log backups that were accomplished after the last full (or differential database backup if accomplished). The main point to remember is that transaction logs should be restored after the last full or differential backup. A key advantage to transaction log restores is the ability to stop a restore at a specific point in time and/or a transaction group label – this cannot be done with database restores.

Filegroup Backup – Backs up the selected database files in the Primary or Secondary filegroups. As introduced in the filegroup section above, SQL allows you to divide up a database into multiple files. In addition to the performance advantages, this allows you to tailor the frequency of backup to the data being backed up – e.g., static data could be backed up much less frequently than dynamic data. However, you should fully understand your database’s topology (what data is in which file and filegroup) so you understand your risk if you implement multiple schedules for certain files. Filegroup backups are followed by transaction log backups and an occasional full database backup (for a full database backup image).
• Restore Impact: While performing a restore of all files in a database’s filegroups is equivalent to a full database restore in SQL 2000 or SQL version 7, you must have a full database backup on hand to restore first (not a constraint in SQL 2000). If you are restoring specific data files (e.g., not all), then you must understand the impact of the restored data on the existing tables before you start the restore. After any filegroup restores, you must bring all filegroups to the same point in time by restoring the transaction logs which followed the filegroup backup(s) that you restored.

The SQL Database backup scheme that will work best for you is based on the size of your environment; the number of transactions processed each day, and the expectations of your users when a recovery is required. To decide which database backup methods to use, consider the following:

• **Small Database Environments** – With relatively small numbers of transactions, consider running a daily full database backup every evening and daily transaction log backups. An alternative would be to use the Simple recovery model in SQL 2000 (or setting the truncate on checkpoint database option in SQL7) and perform a daily full database backup in the evening. Setting the database to this mode causes SQL Server to automatically maintain the transaction log. The downside is the limitation to perform transaction log backups and thus lose the ability to restore to a point in time. This method will remove transaction log management and it will keep your backups simple. If you would prefer the option of not losing a whole day’s work, consider adding a differential database backup during lunch or some other time when the database isn’t busy.

• **Medium Database Environments** – Consider running a weekly full database backup, daily differential database backups, and transaction log backups every few hours. Since most large companies have defined requirements on mission critical backups and restores, your schedule of backups will depend on your requirements. You may need to backup much more or less frequently depending on your SLA.

• **Large Database Environments** – Consider dividing your database into multiple files within a filegroup (or two) using the ideas in the Filegroup section above. Try to backup all of the database’s files in the smallest time window possible to maintain consistency between the files. Perform log backups at least once each day, some environments backup logs every ten minutes. If your database architecture allows you to backup some data very infrequently while concentrating your daily backups on dynamic data, then it is very important that you keep track of which tapes comprise an entire database backup for DR purposes. You might consider performing a full database backup periodically to ensure you have a consistent backup of the entire database.

**Backup Exec Advantage**

In addition to fully supporting all of the database methods above, Backup Exec offers the following advantages:

• **Backup Exec allows an administrator to easily view and select this data, along with any other data types into one schedulable backup job. This gives you the flexibility of managing jobs per server (e.g., Windows OS, SQL, and Exchange in one backup) or by application (e.g., just SQL backups across servers).**

• **Backup Exec includes ease-of-use features that simplify SQL restores. If you are selecting a full, differential, and several log backups to restore, Backup Exec will automatically apply the restores in the correct sequence and bring the database online.**

• **Backup Exec offers industry unique “Guide Me” wizards to help the user determine which SQL backup method is best.**

Backup Exec offers several restore options to suit your needs. These include the ability to: redirect restores to a different SQL instance, SQL Server or Database name; recover to specific log group label and/or date stamp; and recover to various database ready states (warm standby, no recover, or full recovery).
Backup Exec displays SQL Filegroups and allows you to easily select the groups you want to include with each backup.

Options

Database Backup/Restore using Split Mirror Technology – A number of hardware arrays and volume managers support the concept of volume mirroring (RAID level 1) today. Windows Server 2003 also supports volume mirroring using VSS technology. A mirrored volume is simply a real time copy of another volume. Some arrays and volume managers have advanced features that can be manipulated to break off one of the volume copies and mount it on secondary server. This allows a backup to occur on the secondary server without impacting the database server that is still using the original volume. When the backup is finished, the volume can be logically moved back to the original mirror and resynced. This backup method is usually performed in enterprise class data centers that have large, mission critical databases to protect. SQL Server 2000 supports this backup method by providing a way to quickly pause the database so that all files are complete. Only at this time can the mirror be split so that a backup can be successfully completed.

The advantages to a split mirror backup method are:

- The backup becomes a very low resource impact to the SQL Server

- The potential high backup speed since the backup is simply backing up files and not pulling data through the SQL Server API. This can be optimized by using many backup devices to backup the many database files, allowing a parallel backup of all database files at once.
The limitations to a split mirror backup method are:

- Complexity – It is another backup type to use and log backups must still be accomplished. Also, since split-mirror backups occur outside of SQL Server’s Full/Differential backups, you must carefully manage what logs you need to restore when split-mirror restores are accomplished.

- Backups are larger than a full database backup since SQL includes space in the files for database expansion.

**Backup Exec Advantage**

**VERITAS Advantage:** For those customers that need this advanced backup feature, VERITAS offers Storage Foundation for Windows, an enterprise class software solution. This product allows customers to automatically use volume mirroring technology to logically copy SQL databases to another server so that backups can be performed. This offers the advantages of very low impact database backup, offering almost instant recovery, and easier disaster recovery.

**Backup Exec Advantage:** Backup Exec has integrated support for the new ShadowCopy service writers in Windows Server 2003. Backing up SQL Server 2000 databases via ShadowCopy is as easy as clicking on the database you want to backup or restore. The Backup Exec SQL agent extends SQL database protection to include NAS configurations, Full individual filegroup backup, differential database and filegroup backup, transaction log backup, transaction log- no truncate backup, advanced transaction log backup options (no recover and standby) plus automatic consistency check before and after backup.

The Backup Exec SQL agent also extends SQL recovery to include individual filegroup restore, automatic master database restore, automatic alternate drive restore, automatic point-in-time log restore including named transactions, read-only recovery support, automatic restore of deleted databases, automatic consistency check after restore and redirected application restore which includes moving data files to specified volumes.

All of these Backup Exec features are available for both SQL Server 2000 and SQL Server 7.0 configurations using legacy backup APIs.

**Using the Backup Exec SQL agent to protect SQL Server is recommended when:**
- A comprehensive data protection scheme is required (full, differential, transaction log, and filegroup backup) OR
- SQL Server is configured in a cluster OR
- SQL Server is configured using NAS OR
- The database is large OR
- The database is highly active (queries and transactions)

**Using the SQL Writer to protect SQL Server 2000 on Windows Server 2003 is appropriate when:**
- Full backups only are required and
- The database is configured using the simple recovery model and
- Not configured in a cluster and
- Not configured using NAS and
- The database is small and not highly active (queries and transactions)

Note: Intermixing of Backup Exec SQL agent differential, transaction log and filegroup backup with SQL writer backups in a SQL Server 2000 protection scheme on Windows Server 2003 is neither recommended nor supported.
Deployment Guidelines

• Try to coordinate the database backups with the Window Operating System and SQL Server System Database backups so you have a fairly consistent set of data within a small time window. This guideline will help you during disaster recovery.

• Database Consistency Checks – SQL Server offers several types of checks to ensure a database is consistent and healthy. See below for a brief description of the checks and when they should be run, but it is recommended that you at least run the Physical Only check before each database backup to ensure the copy you have backed up is valid.
  
  • Full Consistency check, including indexes. This check will have significant impact on SQL performance; therefore, it should be performed in off-peak hours.
  
  • Full Consistency check with no index check. While not as thorough as a full consistency check that includes indexes, this check is faster and can be done during peak hours with little impact on system performance.
  
  • Physical only check (SQL 2000 only). This low-overhead check method verifies the integrity of the physical structure of the page and record headers, and the consistency between the pages’ object ID and index ID and the allocation structures. This fast check finds most of the common database consistency problems.

• If the SQL database option “Select into/bulkcopy” is enabled, it will allow transactions to enter the database without being entered in the transaction log. Nonlogged operations break the sequence of transaction log backups. The restore of a database using database and transaction log backups is successful only if there is an unbroken sequence of transaction log backups after the last database or differential backup. If you have enabled this option, you should run a database or differential backup and then start running log backups again to save any changes necessary to restore the database.

Backup Exec Advantage

Backup Exec allows you to easily integrate file backups with database backups, so you can better maintain sets of data for disaster recovery preparation. In addition, Backup Exec gives you complete control over SQL database consistency checks – you can select what type of check you want to do before or after the backup or restore. If the check fails before backup, you have the choice of stopping the backup or continuing – either way, Backup Exec will log or alert the error if you desire.
Backup Exec gives you full control of SQL options and even provides a wizard to guide you through the process.
ADDITIONAL MICROSOFT SQL SOLUTIONS FROM VERITAS SOFTWARE

Backup Exec is just one of the VERITAS solutions which support SQL Server. VERITAS develops and sells several solutions that keep SQL Server available (clustering, replication, snapshot management) and backed up (Backup Exec and NetBackup). See below for a list of these products…

• VERITAS NetBackup™ - delivers mainframe-class data protection for the largest UNIX, Windows and NetWare enterprise environments, especially for corporate data centers. VERITAS NetBackup DataCenter provides the most advanced media management available, including dynamic tape sharing, and offers optional database agents like SQL Server, to enable online, nondisruptive backup of mission critical applications.

• VERITAS Storage Foundation for Windows™ (advanced volume management technology for Windows Server 2003) and VERITAS FlashSnap™ for Windows – is for organizations that require uninterrupted and consistent access to mission critical data. VERITAS allows system administrators to more efficiently manage storage environments by virtualizing storage with logical volume management. Logical volume management removes physical limitation of storage, enabling administrators to build higher performance, highly available storage configurations. Once virtualized, the storage can be managed in a more flexible manner, enabling it to be kept online during many of the operations in which the server previously had to be taken offline thus simplifying disk administration tasks for reduced cost of ownership. Storage Foundation for Windows eliminates planned and unplanned downtime, ensures quick recovery from failures, optimized storage I/O performance and protects current storage investments while also allowing freedom of choice for future storage hardware investments.

• VERITAS Cluster Server - is the industry's leading open systems clustering solution which eliminates both planned and unplanned downtime, facilitates server consolidation, and effectively manages a wide range of applications, including SQL Server, in heterogeneous environments. Supporting up to 32 nodes, VERITAS Cluster Server features the power and flexibility to protect everything from a single critical database instance, to very large multi-application clusters in networked storage environments.

• VERITAS Storage Replicator™ - delivers automatic, real-time data replication to Microsoft Windows NT, Windows 2000 and Windows Server 2003 family of products. Whether needed for real-time disaster protection or many-to-one backup centralization, VERITAS Storage Replicator handles even the most demanding replication jobs on the Windows NT and Windows 2000 platforms.
SUMMARY

Like many enterprise database solutions, there are several methods of backing up SQL Server’s data, which can make the administration of the backup process very complex. To tackle this problem, you need a data protection plan and select a reliable backup product that suits your environment. The data protection plan should include the following steps:

1. Determine your SQL Server Service Level Agreement (SLA) needs
2. Research the SQL Server solutions and determine which best suit the needs in your SLA
3. Create a data protection plan which outlines how the solutions will work with your plan
4. Implement the plan and closely monitor the results

This white paper only covered the high-level considerations a SQL Server data protection plan and how you should implement the backup solution. Since SQL Server implementations can scale to very large and complex installations, you may need to consider consulting services to ensure that your implementation is scalable and can be easily recovered in case of disaster.

Regardless of the size or complexity of your SQL Server, the VERITAS Backup Exec™ 9.1 Agent for Microsoft SQL Server offers a highly reliable and easy solution to protect your data. When disaster strikes, the Backup Exec Intelligent Disaster Recovery solution can help get your SQL Server back up and running fast. When fast is not fast enough, VERITAS offers several other solutions to keep your SQL Server available at a higher state than restore utilities can offer.