

# Complete Online Microsoft SQL Server Data Protection

## VERITAS BACKUP EXEC™ 10 *FOR WINDOWS SERVERS*

### Agent for Microsoft SQL Server

SQL Server 7.0  
SQL Server 2000



## TABLE OF CONTENTS

Executive Summary .....	3
Usability .....	3
Reliability .....	4
Why Protect Microsoft SQL Server? .....	4
Why Do you Need the Backup Exec Agent for Microsoft SQL Server? .....	4
Protecting SQL Server .....	5
Introduction .....	5
Application Protection.....	6
Business Requirements.....	6
Options.....	7
Protecting Clustered SQL Servers .....	7
Deployment Guidelines.....	7
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.....	16
Additional Microsoft SQL Solutions from VERITAS Software.....	18
Summary .....	19

## 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 agent will give SQL Server 7 and SQL Server 2000 users complete and customizable protection down to the individual table space or file group.

Is your backup window too small for a full backup? This agent can also perform differential backups as well as Transaction Log backups with automatic truncation. (To completely resolve backup window issues, consider using off-host backup, which is detailed in the Advanced Disk-Based Backup white paper.) Restoring to another SQL Server machine is easy, because Backup Exec Agent for Microsoft SQL Server can redirect a restore. The agent supports rollback and single-pass restores, so administrators can restore databases based on a point in time, rather than a specific backup job. Backup Exec leverages Microsoft's Virtual Device Interface (VDI) to give users the easiest and fastest way available for complete SQL database protection.

## KEY BENEFITS

- Supports both 32-bit and 64-bit SQL Server installations
- Helps maintain the integrity of vital SQL Server data
- Increases the chance of data recovery and minimizes data loss without inhibiting daily database activity
- Recovers the database to a point in time or last commit point, allowing for quick and reliable data restores

## WHAT'S NEW

- Added cluster support with VERITAS Cluster Server
- New restore option called "Force Restore" to improve restore performance by immediately disconnecting any active users, eliminating administrative intervention previously required to disconnect users when performing a restore.
- Support of VSS Snapshot backups, which offer quicker restores

## PRODUCT HIGHLIGHTS

### For SQL Server 2000

- Data recovery to named Transaction Log Marks within the transaction log, so administrators can restore data up to the point at which the data had last been committed to the database.
- Modeling of SQL database backups that can be targeted to fit the individual needs of the business by performing full or differential backups and restores of the file group.
- Expanded data protection parameters that 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, which greatly enhances backup speeds without sacrificing backup accuracy.

### Usability

- Transparent integration online or with "hot" SQL Server backups within regularly scheduled network protection routines.
- Individual table space or individual file-group backup and restore.

- Support for Microsoft SQL Server rollback restores to a specific point in time, rather than a specific backup job.
- Flexible backup launch options for SQL Server, so backup jobs can be launched immediately or on a schedule.

### Reliability

- Use of native SQL Server APIs for both backups and restores, assuring reliable and consistent SQL Server protection.
- Integration 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.

## WHY PROTECT MICROSOFT SQL SERVER?

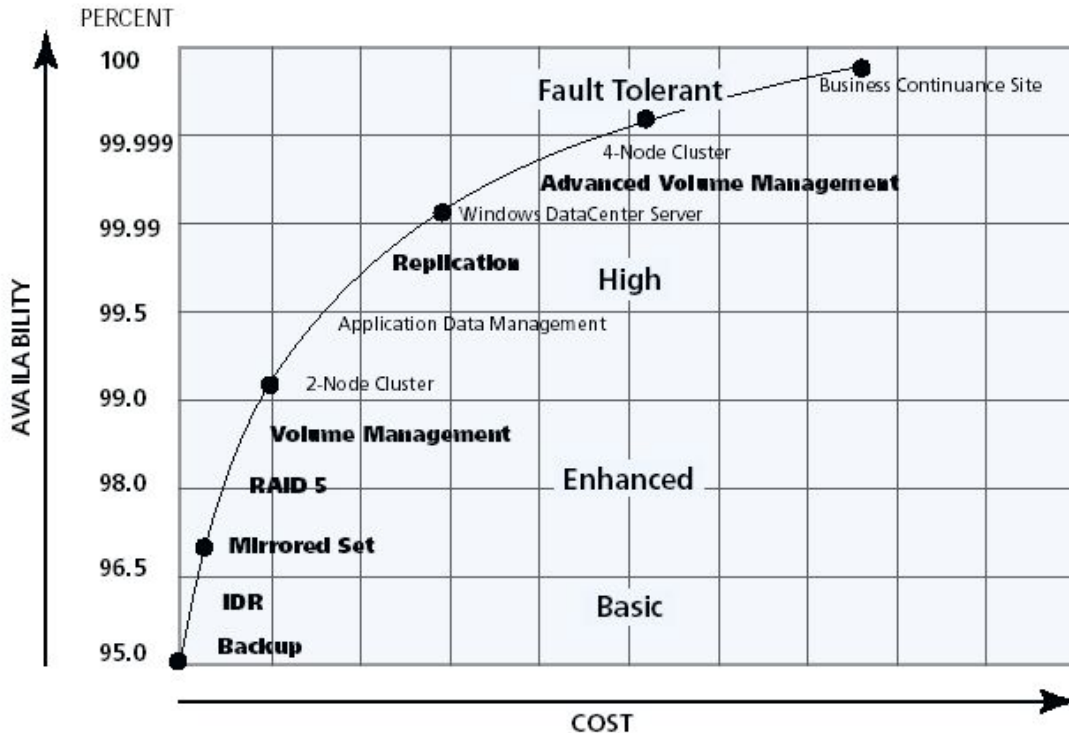
Microsoft SQL Server™ is a general-purpose relational database server that can scale from hosting simple databases to supporting clustered, mission-critical business applications such as SAP. In fact, SQL Server is the most popular relational database on Microsoft Windows, with a 47.4 percent market share (Gartner, May 2004). Simply put, the more your business depends on SQL Server, the more important it is to protect SQL Server.

To maintain Microsoft SQL Server's availability and protect its databases, you need a working and thoroughly tested data-protection and recovery plan, as well as reliable data-protection software. Together, they can ensure the recovery of the SQL Server environment and its databases. The key objectives are 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 addresses several aspects of an SQL Server data-protection plan, focusing on how VERITAS Backup Exec 10.0 *for Windows Servers* and the Backup Exec Agent for Microsoft SQL Server can meet the needs of this plan. It also introduces several other VERITAS products that enhance SQL Server data protection and availability.

## WHY DO YOU NEED THE BACKUP EXEC AGENT FOR MICROSOFT SQL SERVER?

Protecting a database server such as Microsoft SQL Server requires careful thought and planning to meet the availability needs of your company and its budget. The most common method of formalizing these needs is through service level agreements (SLAs). These agreements are contracts between the users and providers (such as the IT department) that 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 SLA.



The basic rule of thumb regarding data protection is the higher the requirement for availability, the higher the cost to achieve that will be. The chart above illustrates this concept and shows the various technology steps 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), giving VERITAS significant experience in this market.

In addition to offering two products (Backup Exec and NetBackup) that support the basic level of availability for SQL Server, VERITAS also offers several products that support your SQL deployment all the way through the highest levels of availability.

## PROTECTING SQL SERVER

### INTRODUCTION

With most database applications like SQL Server, data protection can be divided into two main objectives: (1) preparing for a disaster recovery where all data (the Windows operating system, SQL Server application, and its databases) is destroyed, and (2) preparing for the restoration of all or some of the user database data.

Disaster-recovery preparation is comprehensive and includes protecting the complete SQL application (including the Windows operating system, system state, the SQL Server application directory, and SQL Server's system databases) and SQL Server user databases.

SQL Server provides several ways to deploy and organize user databases and logs, along with several methods to back up and restore them. Each choice can affect the granularity and speed at which you can restore your data, so it is necessary to understand the pros and cons of each.

## APPLICATION PROTECTION

At the application protection level, the focus is to protect the SQL Server's application files and configuration, which includes the SQL Server 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.

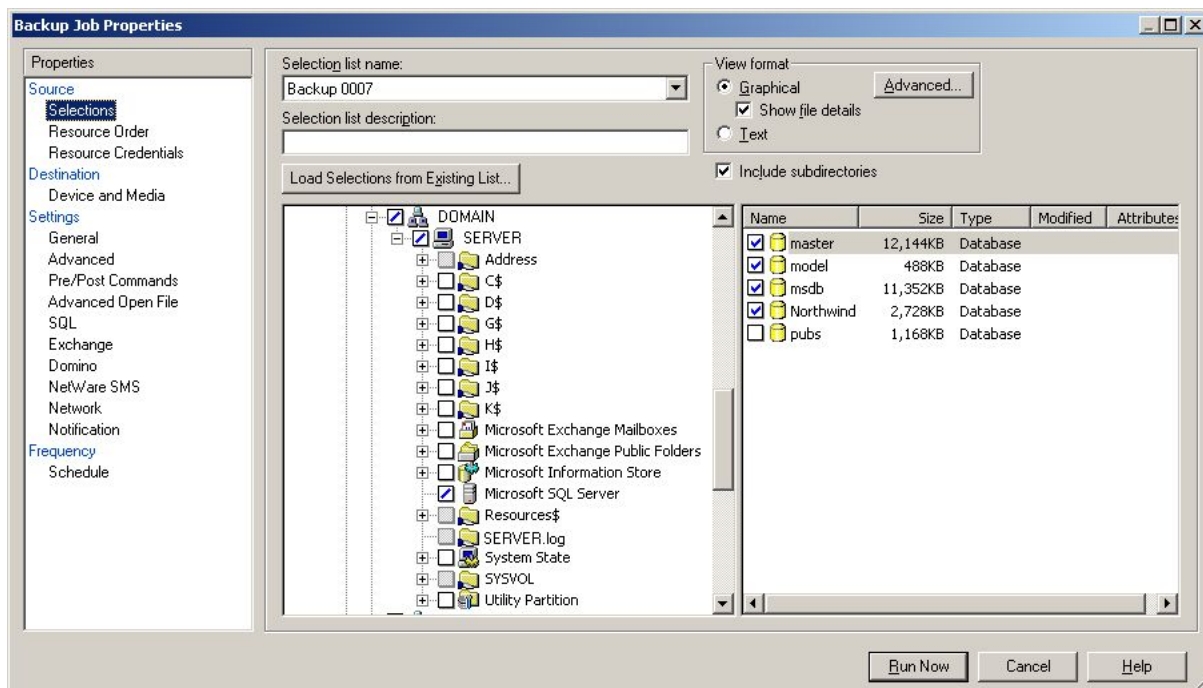
### Business Requirements

**Back up the host server for SQL Server.** Because SQL Server runs on Windows 2000, protecting the underlying Windows operating system and SQL Server's files and settings is very important for quick 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, the Windows system state, SQL Server files, and SQL databases (including the system databases) within a single schedulable job; or you can break these tasks up into multiple jobs, as appropriate for your environment, performance needs, schedule, or data-retention periods. 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 recovering SQL Server.

Backup Exec 10.0 clearly displays all SQL Server data and lets you easily integrate database backups into your backup scheme.



**Back up SQL Server's system databases.** SQL Server itself uses four 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 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. 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 (the distribution database exists only if you setup SQL replication). Like the master database, these system databases can be backed up like any other database. 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 10.0 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: First, recover SQL Server's system databases and then recover the user databases. With Backup Exec, this is reduced to one step: Recover the user databases. Backup Exec backs up the system databases during file backups as offline files (system databases are usually small) and the Intelligent Disaster Recovery Option restores them — so you can 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 clustering technology such as Microsoft Cluster Services (MSCS) and VERITAS Cluster Server (VCS). Clustering technology 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 case one server becomes unavailable. In a cluster configuration, SQL Server presents itself as one virtual server that 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.

### 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 (such as adding or deleting databases, users, stored procedures, or changing database storage). Because 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. 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.

### **Backup Exec Advantage**

Backup Exec fully supports up to a 32-node cluster of SQL on Windows 2000 and Windows Server 2003 (eight is currently the maximum number of nodes that MSCS offers). Backup Exec can automatically restart database backups that were interrupted because of a failover.

## 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 of only a few megabytes to a multi terabyte set of interdependent databases on 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 (SLA)? 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 with 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, 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 can 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 (such as SQL's system databases), doing so is not recommended because then you can restore only to the last full or differential backup.

**File group:** A group of database files. By default, a database belongs to a primary file group, but SQL Server lets you split up a database into multiple files. These files can be organized into multiple secondary file groups that provide the following advantages:

- Increased storage flexibility. File groups let you place specific data (tables) on volumes that can be easily expanded when storage needs arise. This lets you break up a large database into smaller files that can be managed more easily. SQL Server fills all database files in a file group evenly, so eventually all will be the same size.
- Increased performance. File groups let you split a database's files across multiple physical drives, which can increase performance of the database. This also balances 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, rather than restore the entire database.
- Increased availability by isolating database activity. File groups let you place static tables in their own database file, which can then be backed up less frequently than the rest.

## Business Requirements

### Hot (Online) 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 online. The methods depend on the availability and performance needs of the database along with the way you have chosen to configure the database.

### Backup Methods and Their Impact during Restore

**Full database backup.** This 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 — you must make full backups. Full backups are usually followed by differential and/or transaction-log backups.

- Restore impact, Restoring a full database backup takes longer than other backup methods, but once you are finished, the database is ready to be brought online.

**Differential database backup.** This backs up only the changed blocks (extents) within the database since the last full database backup. Because this method backs up only 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 need only 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 two (one full plus one differential) 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 as you can with transaction log backups.

**Transaction-log backup.** This approach backs up the transaction log for the selected database. You can 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 nonetheless are a very efficient way to incrementally back up the database. There are two versions of transaction log backups: truncated and untruncated. The difference is that the truncated version deletes the uncommitted transactions from the log after the log backup is successful, while the untruncated one does not. Normally, you will only use the truncated version, as it is the best way to manage the transaction log's size. The untruncated version is typically only used when the database is corrupted or offline with a problem.

- Restore impact: During a full database disaster recovery, you would normally restore a full database backup, a differential database backup (if one was run since the last full backup), and then any log backups that done after the last full or differential database backup. The key is that transaction logs should be restored after the last full or differential backup, since this lets you stop a restore at a specific point in time and/or a specific transaction group label. (This cannot be done with database restores.)

**File-group backup.** This option backs up the selected database files in the primary or secondary file groups. As noted in the file-group section above, SQL lets you divide a database into multiple files. In addition to its performance advantages, this approach lets you tailor the frequency of backup to the data being backed up. For example, you could back up static data much less frequently than dynamic data. However, you should fully understand your database's topology (what data is in which file and file group) so you understand your risk if you implement multiple schedules for certain files. File-group backups are typically followed by transaction-log backups and an occasional full database backup (for a full database backup image).

- Restore impact: Although performing a restore of all files in a database's file groups 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

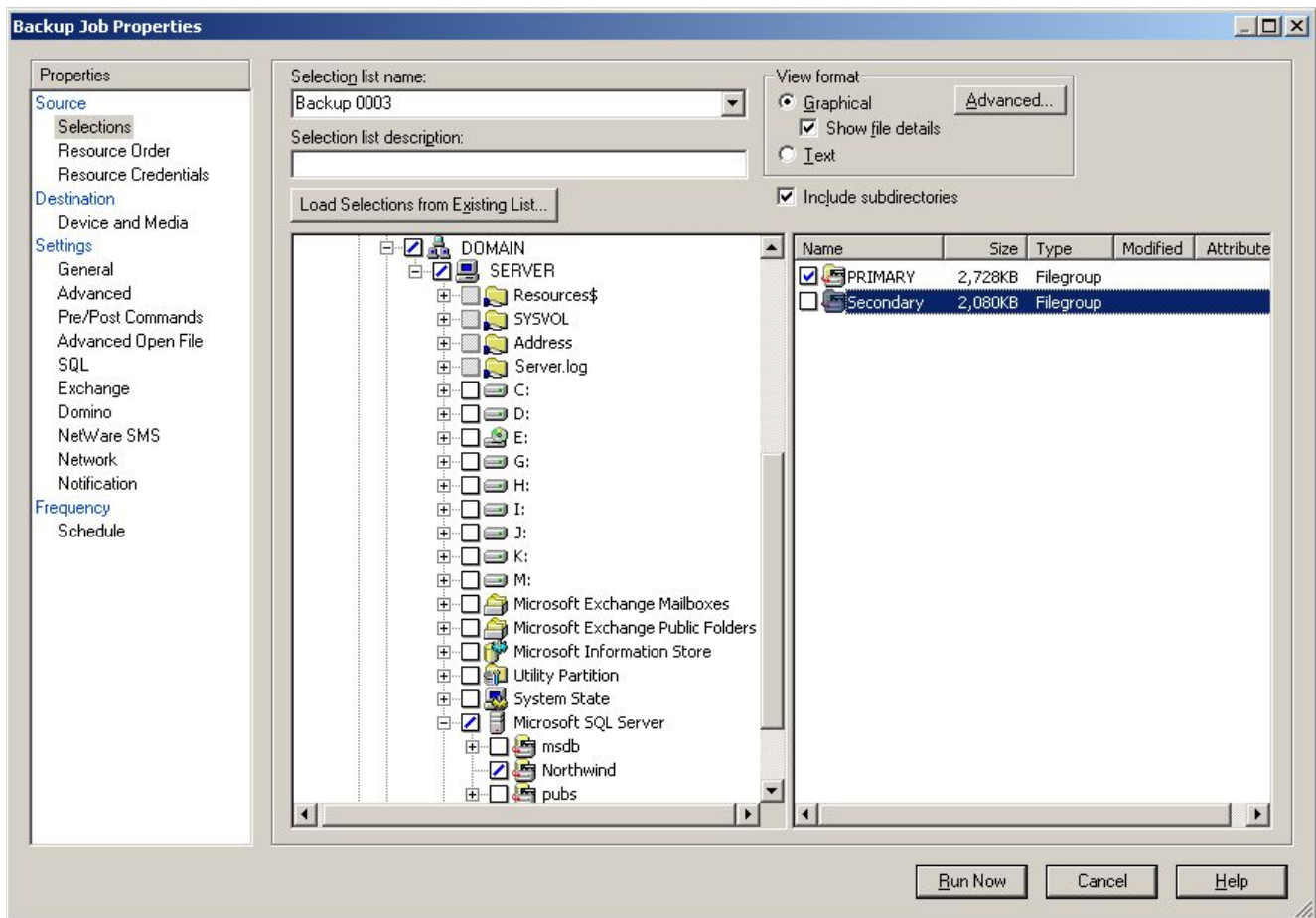
first (except in SQL 2000, which does not require this). If you are restoring just specific data files, you must understand the effect of the restored data on the existing tables before you start the restore. After any file-group restores, you must bring all file groups to the same point in time by restoring the transaction logs that followed the restored file-group backup(s).

**Snapshot backup.** This type of backup requires more disk overhead, and the footprint of snapshot backups are almost always larger than traditional backups.

- **Restore impact:** Restoring from a snapshot backup can cut restore time in half. With traditional restores, SQL must first create a database equal in size to the backed up database and fill the database with zeros. Only then is the actual data restored. Snapshot backups do not require this step.

The SQL database backup scheme that works 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 each evening, as well as daily transaction-log backups. Or use the simple recovery model in SQL 2000 (or set 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 to this approach is that you lose the ability to restore to a specific time. The upside is that this method will remove transaction-log management and 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. Because 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 back up much more or less frequently depending on your SLA.
- **Large database environments.** In large environments, consider several options:
  - Consider dividing your database into multiple files within one or two file groups using the ideas in the file-group section above. Try to backup all of the database's files in the smallest time window possible to maintain consistency between the files. Perform transaction-log backups at least once each day (some environments back up logs every 10 minutes). If your database architecture lets you back up some data very infrequently while concentrating your daily backups on dynamic data, it is critical to keep track of which tapes comprise an entire database backup. You might consider performing a full database backup periodically to ensure you have a consistent backup of the entire database.
  - Consider implementing off-host backup solutions. Off-host backup provides the benefit of creating software- or hardware-based snapshots that can be split from the production SQL server, eliminating any backup window. These snapshots are also mounted on the backup server so you can run a high-speed SAN backup as frequently as desired. (See the "Advanced Disk-Based Backup Option" white paper for complete details.)



Backup Exec displays SQL Filegroups and allows you to easily select the groups you want to include with each

## Backup Exec Advantage

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

- Backup Exec lets an administrator 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 (for example, Windows, SQL, and Exchange in one backup) or by application (for example, 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 transaction-log backups for restoration, 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; recover to various database ready states (warm standby, no recover, or full recovery); or use the new force-restore feature introduced in this release.

### Options

**Off-host backup.** Several 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 lets a backup occur on the secondary server without affecting 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 resynchronized. 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 quickly pausing the database so that all files are complete. Only at this time can the mirror be split to create a successful backup.

The advantages to a split mirror backup method are:

- Elimination of the backup-window problems common to large backups
- Significant reduction of resource usage of the SQL Server
- Potential for significant increase in backup speed, since the backup is simply backing up files and not pulling data through the SQL Server API. This speed can be further increased by using many backup devices to back up the many database files, allowing a parallel backup of all database files.

The limitations to a split mirror backup method are:

- Increased complexity, since it is yet another backup type to use, and log backups must still be performed. Also, since split-mirror backups occur outside of SQL Server's full and differential backups, you must carefully manage what logs you need to restore when split-mirror restores are done.
- Larger backup sizes compared to 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, the VERITAS Backup Exec Advanced Disk-Based Backup Option (ADBO) leverages Storage Foundation for Windows FlashSnap and hardware snapshot providers. These products let customers automatically use volume-mirroring technology to logically copy SQL databases to another server to allow backups. The result is a very-low-impact database backup, with 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 or restoring SQL Server 2000 databases via ShadowCopy is as easy as clicking on the desired database. The Backup Exec SQL agent extends SQL database protection to include NAS configurations, full individual file-group backup, differential database and file-group backup, transaction-log backup, untruncated transaction-log backup, advanced transaction-log backup options (such as no-recover and standby), and automatic consistency check before and after backup.

The Backup Exec SQL agent also extends SQL recovery to include individual file-group 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 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 file-group 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 (in terms of 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
- The database is not configured in a cluster, AND
- The database is not configured using NAS, AND
- The database is small and not highly active (in terms of queries and transactions)

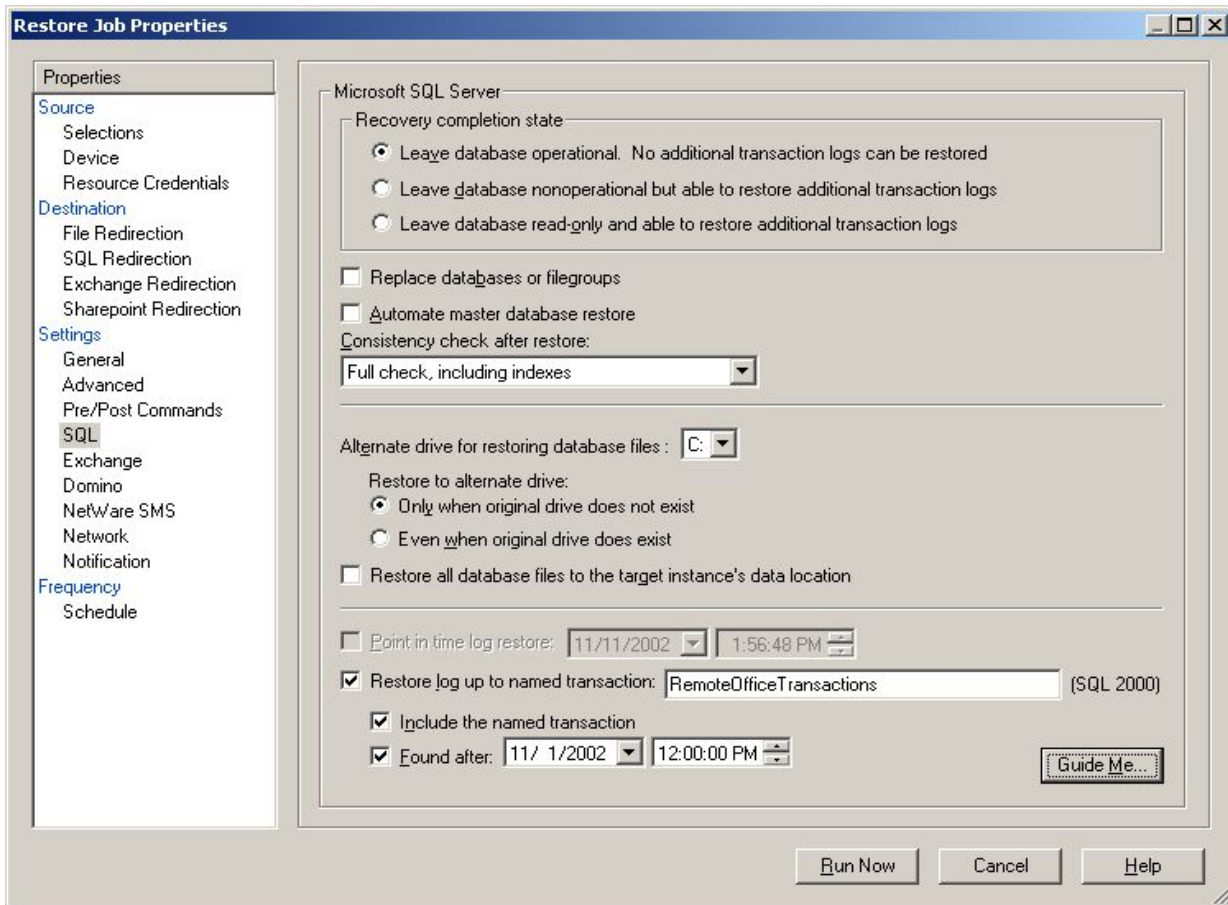
**Note:** Intermixing of Backup Exec SQL agent differential, transaction-log, and file-group backup with SQL writer backups in a SQL Server 2000 protection scheme on Windows Server 2003 is neither recommended nor supported.

## 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. Following this guideline will help you during disaster recovery.
- Be sure to run database consistency checks. SQL Server offers several types of checks to ensure that a database is consistent and healthy. You should at least run the physical-only check before each database backup to ensure the copy you have backed up is valid. Your options include:
  - Full consistency check, including indexes. These check will significant slow SQL performance, so it should be performed during 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 (available in SQL 2000 only). This low-overhead check method verifies the integrity of the physical structure of the page and record headers, as well as 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.
- Disable the SQL database option “Select into/bulkcopy” so transactions can't be entered the database without being entered in the transaction log. Non-logged operations break the sequence of transaction-log backups, and database restoration using database and transaction log backups will be 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 lets you 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 can stop the backup or continue; 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 that support SQL Server. Other solutions keep SQL Server available (via clustering, replication, and snapshot management) and backed up (via Backup Exec and NetBackup).

- **VERITAS Backup Exec Advanced Disk-based Backup Option.** This option, when used with the Backup Exec Exchange Agent, can dramatically improve your overall Exchange server data-protection strategy. It lets administrators create a mirror of their Exchange data, break off the mirror, mount it on their backup server (thus backing up locally) and then resynch the mirror with the Exchange server at the end of the backup. This powerful feature eliminates dependencies on backup windows without slowing your Exchange servers during backup.
- **VERITAS Storage Replication™.** This delivers automatic, real-time data replication to the Microsoft Windows NT, Windows 2000, and Windows Server 2003 family of products. Whether needed for real-time disaster protection or for many-to-one backup centralization, VERITAS Storage Replicator handles even the most demanding replication jobs on the Windows NT and Windows 2000 platforms.
- **VERITAS NetBackup™.** This 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, non-disruptive backup of mission-critical applications.
- **VERITAS Storage Foundation for Windows™ (advanced volume management technology for Windows Server 2003) and VERITAS FlashSnap™ for Windows.** These are for organizations that require uninterrupted and consistent access to mission-critical data. VERITAS lets system administrators more efficiently manage storage environments by virtualizing storage with logical volume management. Logical volume management removes physical limitation of storage, so administrators can build higher-performance, highly available storage configurations. Once virtualized, the storage can be managed more flexibly, so it can be kept online during many of the operations in which the server previously had to be taken offline. This greatly simplifies disk-administration tasks, reducing 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 providing freedom of choice for future storage hardware investments.
- **VERITAS Cluster Server.** This is the industry's leading open-systems clustering solution. It 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 has the power and flexibility to protect everything from a single critical database instance to very large multi-application clusters in networked storage environments.

## SUMMARY

Like many enterprise database solutions, there are several methods of backing up SQL Server's data. Having such a range of options can make the administration of the backup process very complex. That's why you need to create a data-protection plan and select a reliable backup product suited to 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 defined in your SLA.
3. Create a data-protection plan that outlines how the solutions will work with your plan.
4. Implement the plan and closely monitor the results.

Because 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 10 Agent for Microsoft SQL Server offers a highly reliable and easy-to-use 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 the 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.

**VERITAS Software Corporation**  
Corporate Headquarters  
350 Ellis Street  
Mountain View, CA 94043  
650-527-8000 or 866-837-4827

For additional information about VERITAS Software, its products, or the location of an office near you, please call our corporate headquarters or visit our Web site at [www.veritas.com](http://www.veritas.com).