

# Backup Scenarios for Successful SQL Server Restores and Recovery

## Introduction

The intent of this document is to explore various backup scenario plans for SQL servers and come up with a recommendation to successfully restore them when the need arises. Database Administrators (DBAs) typically spend most of their time working on configuring and monitoring backup plans, but rarely take adequate time to test the restore path. DBAs should ask how confident they are that the backed up data can be restored when the need arises. The typical problem faced by DBAs is that they have to handle an ocean of backup sets and don't know where to begin. More often, they spend hours trying to restore but end up restarting different series of backup files. All these lead to more downtime and data loss, in spite of having access to successful backups.

## Backup Strategy

Taking backups is just the starting point for securing databases. What is crucial is the ability to anticipate disasters that can occur in a typical corporate environment. This list includes, but is not limited to, catastrophic events, disk crashes, network malfunctions, malicious data modifications, database corruptions, accidental data updates and so on. Each time the DBA might want to pick up a different restore sequence best suited for limited or no data loss with minimal downtime.

Having all these disaster types in mind, it is always challenging for DBAs to architect the backup plans and scheduled times, which will have a minimal impact on the production systems. In addition, backup plans need to accommodate future growth in the size of databases, as increase size requires increased backup time. It is unacceptable to have a backup plan in which by the time last night backup completes, next day's backup starts again and in such a scenario the system always would be in the backup mode. In the subsequent paragraphs we will provide vital recommendations on backup plans, disaster simulations and restore paths for various situations and different types of databases including System and user databases.

To choose the most optimal backup plan, one must answer the following questions:

1. How much data can we afford to lose when disaster occurs?
2. What is the acceptable downtime window for restore and recovery?
3. Is this a 24x7 database?
4. How large is the database and what type?
  - a. Is it a Static database? Loaded only once?

- b. Is it a read-only database with periodic updates?
- c. Is it a database that is frequently updated?

Here are some backup recommendations for SQL system databases:

### **Master Database:**

The master database is essential for all Microsoft SQL Server operations. If the master database fails or becomes corrupt, it takes the whole server down with it. Unlike a critical user database that handles real-time customer transactions, the master database need not be backed up every hour or every half-hour. One needs to backup the master database for the following reasons:

- a. after it is created
- b. if the configuration values are changed
- c. if the SQL logons are configured
- d. any changes are made to the database

### **Model Database:**

The Model databases a system database used in new database creation and is used as a template. There is no specific reason to backup the model database very often. Whenever a new object is created, database configuration is changed to make sure that all new databases are created with those objects and configuration. In such a situation, the database has to be backed up. Making a complete database backup is, in most cases, good enough.

### **MSDB Database:**

The job schedule and the history are maintained and stored in the MSDB database. Hence, whenever jobs are scheduled through the SQL Server Agent, MSDB should be backed up. Also the information about replication and log shipping is kept in this database. Whenever, changes to a configuration related to the above mentioned items are made, the database should be backed up.

### **Distribution Database:**

The Distribution database is available when replication is configured and the server is acting as a distributor. The recommendation is to schedule complete backups after snapshots. In the case of transactional replication, it is advisable to schedule regular log backups.

### **Tempdb**

Normally, Tempdb need not be backed up. This database is recreated each time the SQL Server is started.

So far we have looked at backup recommendations for system databases. For user-created databases, let's examine the following scenarios.

#### Scenario 1:

Consider a non 24x7 Database (about 500 MB to 1GB)

You should perform a complete database backup every night during the non-operational window and do transactional backups every hour. This approach provides both a point-in-time restore and a point-of-failure restore.

#### Scenario 2:

Consider a 24x7 Database (about 500 MB to 1GB)

You should perform a complete database backup every night during non-peak hours followed by a differential backup during off-peak time of the day. Transactional backups should be performed every 15 minutes, round the clock. This approach provides both a point-in-time restore and a point-of-failure restore. The restore time is also considerably less.

#### Scenario 3:

Consider a 24x7 Database (about 1 GB to 10GB)

You should perform a complete database backup every night. Do differential backups during off-peak time of the day. Do transactional backups every 15 minutes round the clock. Depending on the activity in the system, change transactional frequency.

#### Scenario 4:

Consider a 24x7 Database (about 10 GB to 60GB)

You should perform a complete database backup every night. Do differential backups during off-peak time of the day. Do transactional backups every 15 minutes round the clock. Depending on the activity in the system, change transactional frequency.

If the database grows beyond 60 GB, the time it takes to do a complete backup becomes the key factor. Taking individual file/file group backups is highly recommended thereafter. Today, sophisticated RAID disk systems are available and you should look at multiple files/file groups from a flexibility perspective rather than from a performance angle.

File/File Group backups give greater flexibility to handle very large databases for a number of reasons. Let's take a look at some of them.

- Recovery from isolated failures is faster. Only the damaged file needs to be restored rather than the entire database.
- Enables greater flexibility to schedule backup jobs for very larger databases where the complete database backup is almost unmanageable.
- One can organize file groups based on the updatability for better backup strategies. This enables frequent backups to frequently modified data files, rather than repetitive backup of static portions of the database.

You should be aware that these file/file group backups are not complete without transaction log backups. When restoring individual files, the transactions have to be rolled forward to make sure that the file or file group is consistent with the rest of the database. In order to reduce the number of transaction logs to be rolled forward, one should look into doing differential backups.

The primary disadvantage of the file and file group backups is the complexities involved in maintaining a complete set of backups to make sure that the restore yields a consistent database at the end.

## **Summary**

In conclusion, one should carefully think through their backup strategies before embarking on a backup plan. We looked at some of the best practices for backing up SQL system databases. Also, the article discussed through various scenarios the backup strategies for databases of varying size and criticality. DBAs should take into consideration potential disasters and execute their backup plans to minimize loss of time and resources. By looking at the larger picture and taking into consideration all eventualities, one will be able to minimize downtime and improve productivity.

## **About Sonasoft™**

Sonasoft is a leading provider of disaster recovery and business continuance solutions that simplify and lower the cost for mission critical multi-server systems. Founded in San Jose, CA, in 2002, Sonasoft provides integrated hardware and software based solutions that simplify and automate the process of database backup and recovery, centralize the management of multiple servers, and cost-effectively provide a disaster recovery strategy to protect valuable data. SonaSafe™ is a plug'n'play appliance that is specifically designed for disk-to-disk backup and recovery. SonaSafe appliance reduces TCO and provides faster ROI compared to tape and other solutions available in the market. For more information, please visit [www.sonasoft.com](http://www.sonasoft.com)