General Backup Strategies

Computers in general are very reliable. You may run your system for months or even years without experiencing any problems that cause you to lose information on your system. But businesses are more and more dependent on computers and the information that is stored in them. The information that is in your computer may not be available anywhere else. So every system needs to back up and restore some or all of its data. There are numerous backup strategies a company can use. In the following, you will find a short introduction to the concepts of backup, restore and recover for databases in general and Tamino in particular. The following topics are covered:

Concepts

In the following, you will find explanations of general notions and terms with regard to backing up databases. Most of them are available in Tamino, unless mentioned otherwise.

Backup

A database is saved to one or more output devices. Note that the term "backup" is used both for the process of saving the data and for the resulting data sets. Making a backup should be possible online, parallel to normal database update activities, and save a transaction-consistent state of the database. Backups should be done at a time when there is a low data load. Several backup concepts are conceivable:

Online backup: A backup during a normal update database session.

Offline backup: A backup when database updates are disabled (the server is down, or in stand-by mode, or in read-only mode).

Complete backup: All data of the whole database or the logical or physical subset of the database is saved.

Incremental backup: Only the data which has been changed since the previous backup is saved. A recovery is only possible if a previous full backup is available, as well as all following incremental backups. Incremental backup and recover operations are fast and efficient.

Full backup: The complete database is saved.

Partial backup: Only a logical or physical subset of the database is saved.

Note:
Please note that partial backups are not available in Tamino.

Restore

A restore *recreates* the content of the database at backup time from the backup devices.

Full restore: The complete database is restored. A full restore is only possible after a full backup.

Partial restore: Only a logical or physical subset of a database is restored. A partial restore is possible after a full or partial backup (but not available in Tamino).

Recover from Database Logging

A database backup alone allows you only to restore one state of the database as it was at backup time. After a data failure, however, it may not be sufficient to recreate that state of a database, but the state of the database just before the failure occurred. For this reason, all update operations are logged on log spaces.

After the database has been restored, the log spaces are read and the logged update operations are repeated, so that the database is returned to the state that was valid at the time when the last log entry had been created. This process is the recover process.

Full recover: The complete database is recovered. A full recover is only possible after a full restore.

Partial recover: Only a logical subset of the database is recovered. A partial recover is possible after a full or partial restore (but not available in Tamino).

Non-Parallel Backup and Restore/Recover

Normally, a backup is performed in non-parallel mode: The data blocks are written in one stream to the backup devices, or read in one stream from the backup devices.

Parallel Backup and Restore/Recover

A parallel backup writes in parallel to more than one backup device; a parallel restore reads in parallel from more than one backup device. This increases the speed of the backup/restore process, if the backup device is much slower than the disks where the database is stored.

Replication

Conceptually different from a restore process, a database can be *created* from backup. In this case a new (duplicate) database is created with the content of the database at backup time.

Changes to the database that occur after the creation from backup can then be replicated in a replication database. Unlike a conventional backup that is restored from tape or CD, the replicated database is available to applications as soon as they can be pointed to it. For further information, see the Replication Guide.

Requirements for Backup, Logging, and Restore/Recover

After having mentioned the basic notions of backing up, the question arises why we need backup, restore and recover at all. The simple answer to that is that you want to be able to recreate a previous state of a database after an error has occurred. The reasons for errors are manifold and are dealt with in detail in the next section Recovery from Data Failures. Let us first consider a few requirements for being able to recreate a previous state of a database.

Database Synchronization

When a backup is performed online, it is important to be able to create a consistent state of the database after the corresponding restore. To achieve this, in Tamino a database synchronization is performed at the end of the backup: New transactions are postponed until all open transactions are finished. When all updated blocks have been written to disk, the database is in a consistent state. When this state of all database blocks is contained in the backup, it is possible to restore this (consistent) state of the database.

The restore operation recreates this consistent state of the database. Note that the restored database must be logically identical, but may be physically different. For example, the restore operation can defragment the data.

When the database server is active, it logs all update operations in the database log. After you have restored the database, the recover operation reads the logs and repeats all update operations which have been performed until the required timestamp or until the end of the logs.

Other Requirements

- If you want to be able to perform a recover after a disk error, it is necessary that the database logs and the backups are not on the same disk. This is not necessary if you only want to be able to recover from a software failure, because you have a hardware solution which guarantees that the database is not destroyed because of a hardware failure.
- Disaster Recovery (based on restore/recover) requires that the backup and log spaces be physically copied to a new computer center. For example, if the current log space is copied only after it has been closed, you are not able to reapply the changes that occurred during the current server session.

An alternative to performing a backup is just copying the database spaces to another place. But this has some disadvantages: First, it is only allowed if no update session of the database server is active. Otherwise the saved database spaces are inconsistent. Second, Tamino does not know of these “backups”. This means that old log spaces are not deleted and not released by Tamino. In addition, log spaces cannot be applied after a restore. For this reason, it is not recommended to copy the database spaces to another place instead of performing a normal Tamino backup.

Recovery from Data Failures

One of the most important tasks a database administrator has to accomplish is to define for each database how to handle data failures. There basically are four different kinds of data failure which can occur:

- Hardware Errors
- Software Errors

- Disaster
- Second Failure

Hardware Errors

A typical hardware error which may destroy a database is a disk failure. In this case, a Tamino restore/recover is a good possibility to handle the situation (see Internal Backup and Restore in Tamino). If you want to be able to perform a recover after a disk error, it is necessary that the database logs and backups are NOT on the same disk. Normally, a disk error is noticed as soon as it occurs. Hardware errors, that are not immediately recognized, are more problematic, for example if a disk read operation does not display an error, but returns wrong data. This situation is similar to software errors (see next section Software Errors). Another solution for handling disk errors is saving backups on a tape.

There may be other hardware errors which do not require a restore/recover, but for example a new database start to be performed after repairing the hardware. Note that there are also other concepts of handling disk errors, for example RAID 5 or cluster solutions:

- RAID 5 or disk mirroring: The data is stored redundantly on the disks. If a disk is corrupted, the data is automatically read from other disks. The computer operator must only replace the corrupted disk. The data is automatically recreated on the new disk.
- Replication: After a hardware error has occurred, a replication of the database becomes the master database. It must be made available with the name of the original database. The advantage of this solution is that the database is available without time losses required for a restore/recover process. On the other hand, some transactions may be lost in this process.

The following table compares various possibilities available to recover from a hardware error:

Solution	Special Hardware Requirements	Recovery Time	Loss of Data	Remarks
Internal Restore/Recover in Tamino	None	Long	No	-
RAID 5 or disk mirroring	There are hardware or software based solutions, where the operating system manages the disks	None (the user does not notice that there is a disk failure)	No	If the system is not based on physically separated storage devices, an additional recovery solution should be provided in case the whole storage system fails.
Tamino Replication	None	Short, but in contrast to high availability, the replication database must be made available as the	Yes; because the replication is done asynchronously, the last transactions may be lost.	This solution allows also recovery from other hardware errors, for example CPU failures.

Solution	Special Hardware Requirements	Recovery Time	Loss of Data	Remarks
		master database manually.		

Software Errors

Contrary to recovery from hardware errors, an automatic recovery from software and handling errors is not possible. For the system, a software error is like a normal update operation. The database administrator has several possibilities for handling the problem:

- Perform a restore/recover to a state before the error occurred.
- Tamino software error: In some cases it might help to restore a backup created before the erroneous Tamino version was installed, and to recover all logs with a Tamino server in which the problem had been fixed. (Note that it is possible to restore a backup from a former Tamino version, even if that Tamino version is not installed).
- Try to repair the error, for example by updating the corrupted data or unloading the data that is not corrupt, deleting the corrupted data and reloading the correct data.

The solution depends very much on the individual error situation. Nevertheless, it is useful to perform regular backups, so that backup and restore/recover is a feasible possibility in each situation.

Disaster

It may happen that not only part of the hardware is erroneous, but that the whole hardware system is destroyed, even the complete computer center. In this case, it is necessary to make the data available on another computer, in a different place. This scenario is called disaster recovery. Concepts of disaster recovery are not necessarily based on backup and restore mechanisms. You can also use replications or cluster solutions with physically distributed storage devices. In any case all data required for the disaster recovery must be saved at a remote location. The following table shows the various possibilities you have for disaster recovery in Tamino:

Solution for Disaster Recovery	Special Hardware Requirements	Required Recovery Time	Remarks
Tamino Restore/Recover	None	Long	The updates of the current logs are lost if the log spaces are only copied after they have been finished.
Disk mirroring on remote location	Yes, but possibly there are also software-based solutions available	Short, the server needs only to be started on the target machine.	The same precautions as for a cluster solution are necessary.
Tamino Replication	No	Short, but the replication database must be manually made available as the destroyed master database.	Updates may be lost!

For more information about disaster recovery in Tamino, see the section Disaster Recovery in this guide and the documentation about High Availability.

Second Failure

In addition to a single hardware error, the database administrator must be aware of the fact that there is a small risk of a second failure. Standard backup solutions guarantee recovery only if not more than one disk crashes. However, if for example you perform an internal backup, a disk containing a database space has crashed, and the backup is not readable, the complete data is lost.

Depending of the kind of recovery, the following strategies can be provided in case of a second failure:

- If you have a separate solution for disaster recovery, you can use this solution for a second failure. But be aware of the fact that if the only solution for recovery from hardware failures is the disaster recovery solution, this may not be sufficient.
- If you are performing internal backups, you can use a previous backup. In this case, it is important that the different backups are stored on different physical devices. If you want to be able to also perform a recover process in the case of a second failure, you must also save the log spaces to different physical devices. If you restore an older backup, recovery will take longer than usual, because also the logs created between this backup and the backup which is no longer readable must be applied. You can avoid this by copying the backups to another physical location.
- Logs should be copied if you do not want to lose updates because logs are no longer readable.
- If you use RAID technology, you can additionally perform backups and copy the backups to another device. You will usually reduce the frequency of the backups, for example once a week instead of once a day.

Time Considerations

In the day-to-day administration environment, there normally is a requirement stating that the database should be up after a failure within a given amount of time, for example within two hours. This means that the database administrator must estimate the time necessary for a restore/recover process. Total recovery time is the result of summing up the restore time and the recover time. Use the figures given in the following rule-of-thumb example, to calculate an estimate of the restore/recover time.

Assume the estimated update time is half an hour and the estimated recover time for the updates of one day is a quarter of an hour, and that you have 5 working days with update activity. Assume the restore/recover time should be no more than 2 hours, after which you should do a weekly backup. If the failure occurs very shortly after the backup, the restore/recover time would be about half an hour. If the failure occurs after one week, the restore time would be about 0.5 h + 5 * 0.25 h = 1.75 h. In this case, it is recommended to perform a weekly backup. There may be 20% more update activities than usual, and the restore/recover time is still not more than 2 hours.

Note the following rules-of-thumb:

- The restore time is proportional to the backup time. Compare the backup and restore time and use the resulting factor: When the database grows, you can estimate the restore time by multiplying the current backup time by the factor.

- The recover time is proportional to the size of the log files, unless mass loads or index creation operations must be recovered. Also, the recover time for a given amount of log files may vary, depending on the number of update operations.