Chapter 1

High-Availability Options

Database mirroring is one of several high-availability practices or solutions that you can apply in your SQL Server environment. The term high availability refers to the building of fault-tolerant systems to maximize uptime. System failures are going to happen. High availability enables a system to recover from these failures and cover them up from the outside world.

The first step in choosing a high-availability solution for your system is to understand the options available to you. The four options generally employed for high availability are:

- Database mirroring
- Failover clustering
- Log shipping
- Replication

While this book is about mirroring specifically, you should have some idea of the other options. And you should know when to choose mirroring, which is to say that you should be aware of what requirements trigger mirroring as a solution. In addition, you will see that database mirroring works well with other high-availability options to provide even more fault tolerance, leading to maximum uptime for your applications.

High Availability vs. Disaster Recovery

Before we discuss the different high-availability solutions, it is important to understand the difference between high availability and disaster recovery. The terms high availability and disaster recovery are often confused or thought of as the same thing. Just because you have implemented a high-availability solution does not mean that you are prepared for a disaster. High availability generally covers hardware- or system-related failures, while disaster recovery (DR) covers you in the event of a catastrophic failure due to environmental factors. In many cases, database mirroring provides both high availability and disaster recovery by offering a second copy of the database that can be available to your users in a matter of seconds. While some of the high-availability options may help you when designing your DR strategy, there is not one single solution that works for everyone.

The goal of high availability is to provide an uninterrupted user experience with zero data loss, but high availability can have many different meanings depending on who you ask. According to Microsoft, “A high-availability solution masks the effects of a hardware or software failure and maintains the availability of applications so that the perceived downtime for users is minimized.”

Many times users will say they need 100 percent availability, but what does 100 percent availability actually mean to the user? Does being 100 percent available mean that the data is 100 percent available during business hours or does it mean the data is available 24x7 Monday through Friday? High availability is about setting expectations and then living up to them. That’s why one of the most important things to do when dealing with high availability is to define the expectations in a service level agreement (SLA) that can be agreed upon and signed by all parties involved.

Some of the things you should cover in the SLA are maintenance windows, the amount of recovery time allowed to bring the system back online due to a catastrophic failure, and the amount of acceptable data loss, if any. Defining a maintenance window allows you to apply service packs, patches, and upgrades to the system to ensure optimal performance and maintain compliance. Having a maintenance window allows you to do this in a tested and planned fashion. A drop-dead time should be determined so that a back-out plan can be executed if problems are encountered, ensuring system availability by the end of the maintenance window.

Defining the amount of time allowed to recover from a disaster along with the maximum allowed data loss will help you determine what techniques you may need to use to ensure that your SLAs are met. Every organization wants 100 percent availability 100 percent of the time, but when presented with the cost of a system that would even come close to achieving this goal, they are usually willing to negotiate attainable terms. It is important to have an understanding of what it means for a system to be unavailable. Is it a minor inconvenience because users within your organization will not be able to log their time, or are you losing thousands of dollars in revenue every hour the system is down? Answering these kinds of questions will allow you to justify the cost of an appropriate solution. Each high-availability method brings unique characteristics to the table, and unfortunately there is no cookie-cutter solution. In the next few sections, we will discuss the individual techniques used to meet your high-availability needs.

**Database Mirroring**

Database mirroring protects at the database level. You maintain a mirror, or copy, of the live database on a separate instance of SQL Server (preferably on a separate machine). As transactions occur on the live database, the primary instance copies those transactions to the mirror database. The live database will fail over to the mirror if the live database becomes unavailable due to failure of the server, the live instance, or the database itself.

You are required to keep the mirror database in a restoring state, which means it is not available for incoming requests. However, you can create a database snapshot of the mirror database, which provides a point-in-time, read-only view of the database.

Database mirroring offers the following advantages and benefits:

- Protection against database failures
- Automatic failure detection and failover
- Support of easy manual failover
- Automatic client redirection
- Multiple operating modes
- No special hardware requirements
- Minimized chance of data loss
- No single point of failure
- Relatively ease of set up and configuration

Database mirroring is highly configurable and easy to work with. It can operate in synchronous or asynchronous modes, sometimes called high-safety and high-performance modes, respectively. It can operate with or without a witness server, and it can be configured for automatic or manual-only failover. We will provide a more detailed overview of the mechanics of database mirroring in Chapter 2.

**Failover Clustering**

Failover clustering protects an entire instance of SQL Server, whereas database mirroring only protects an individual database. You can cluster two or more identical SQL Server instances together using Microsoft Cluster Service (MSCS). The SQL Server instances share common resources such as a connected disk subsystem, usually a SAN (storage area network.) This cluster of servers presents itself as