25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 691

CHAPTER 19

Administering and Managing Database Mirroring

This chapter takes a systematic look at database mirroring, one of the four high-availability alternatives offered with SQL Server 2005. Database mirroring is probably one of the most anticipated features of SQL Server 2005. Unfortunately, it wasn't ready for prime time and was actually pulled and not officially supported when Microsoft shipped SQL Server 2005, creating slight disappointment. It wasn't until the release of SQL Server 2005 Service Pack 1 that database mirroring was officially supported in production environments.

Database mirroring offers increased database availability and database protection by providing and maintaining a hot standby database on another instance of SQL Server 2005. With database mirroring, continuous support is given to organizations, bolstering operations by decreasing downtime and reducing data loss.

To ensure organizations can reap the full benefits of database mirroring, the topics in this chapter are geared toward giving you the knowledge base necessary to understand the full potential of database mirroring and how it meets different business scenarios as well as how to implement and maintain it successfully. Specifically, the chapter focuses on an overview of database mirroring, terminology, and ways to use database mirroring. The middle sections of the chapter focus on database mirroring configuration and administration. The final sections of the chapter discuss how to manage and monitor database mirroring.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 692

What's New for Database Mirroring with SQL Server Service Pack 2?

Although database mirroring was the biggest buzz with the new features released with SQL Server 2005 Service Pack 1, Microsoft has not released any new database mirroring features with Service Pack 2. However, Microsoft has addressed and bundled all the previously released hotfixes and patches that customers identified when using database mirroring in production. A summary of some of the major bugs follows:

- Prior to the release of Service Pack 2, an error occurred when database administrators manually attempted to fail over the principal database. In many cases, the growth increment of a principal database changed from a fixed type setting to a percentage, causing excessive growth increments of data files and log files. Microsoft released a hotfix to address the problem. Today, this bug is also addressed with Service Pack 2.
- Error 1456 was erroneously logged in the event viewer of the SQL Server when attempting to add witnesses to a new database mirroring session with a database name that was exactly the same as an existing database mirroring session. This error prevented the witness from being added to the new session. This issue has now been addressed.

SQL Server 2005 Database Mirroring Overview

As mentioned in the introduction, database mirroring offers increased database availability and database protection by providing and maintaining a hot standby database on another instance of SQL Server 2005. Its usefulness is best witnessed when a failure takes place on a primary database. In this situation, the standby database becomes active and clients are redirected without the organization experiencing data loss or downtime.

Database mirroring is also commonly used to meet disaster recovery requirements and, therefore, should not be recognized only as an availability mechanism for a local site. When database mirroring becomes an integral part of an organization's disaster recovery plan, a hot or warm standby database is typically placed in a physical location other than the primary active database.

Note

The primary database is commonly referred to as the principal database, and the hot or warm standby is referred to as the mirror database.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 693

The principal database handles client activity, whereas the mirror database receives continuous transaction log changes through a dedicated and secure TCP endpoint. This process keeps the mirror database up to date and ready to take on client operations in the event of a failure. Depending on the configuration/operating mode, database mirroring can be configured for either synchronous or asynchronous operations.

Figure 19.1 depicts the internals of a database mirroring session.

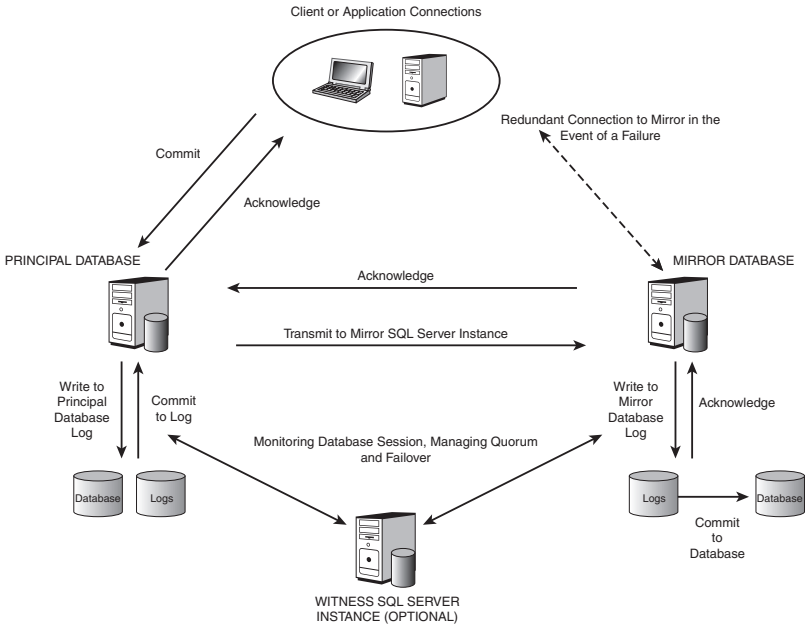

FIGURE 19.1
Overview of database mirroring.

Many database administrators find similarities between database mirroring and log shipping. They often refer to database mirroring as real-time log shipping or log shipping on steroids. However, in database mirroring, unlike log shipping, the primary server does not ship the logs to the standby server based on a time increment. Database mirroring transactions are continuously sent and committed between the principal and mirror; therefore, the databases are kept up to date.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 694

Note

It is not possible to configure a database mirroring session on the SQL Server system databases such as the master, msdb, tempdb, or model databases. Failover clustering would be the alternative to provide high availability on the system databases.

Database Mirroring Terminology

Although you may be eager to install SQL Server 2005 database mirroring, you should take the time to fully understand all the new terminology and components that make up this new high-availability solution. By doing this, you and your organization can avoid running into roadblocks and have an easier time with the installation. To prepare you for the installation, review the following terms regarding database mirroring:

- **Principal Database**—The principal database is the primary server within a database mirroring solution. This server maintains a live copy of the database that is accessed by clients and applications.

Note

The principal database must reside on a separate instance of SQL Server than the mirror database.

- **Mirror Database**—The mirror database is the target database that reflects all the changes of the principal database through a secure dedicated channel. The mirror database is a hot or warm standby and is continuously updated by transferring transaction logs from the principal database in chunks.

Note

The Mirror SQL Server instance does not require a SQL Server license if the server is strictly used as a standby.

- **Witness Server**—The witness server is an optional component within a database mirroring session. Typically, this component resides on a dedicated SQL Server instance independent of both the principal database and the mirroring database servers. The witness facilitates the

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 695

quorum role and monitors the database mirroring session. It initiates a failover from the principal to the mirror database in the event of a disaster. You may view the witness server as a requirement to support automatic failovers between the principal and mirror database instances.

- **Quorum**—The quorum monitors the state of a mirroring session. It controls the failover process based on communication among the principal, mirror, and witness databases. The principal server maintains the primary role of owning the database by maintaining quorum with either the mirror or witness. At least two servers are required to form a quorum; if the principal loses quorum with the mirror and/or the witness, a failover is initiated.
- **Synchronous/Asynchronous**—A database mirroring session can transfer data between the principal database and mirror database by either a synchronous or an asynchronous operation. When you use the synchronous transfer mechanism, a transaction is successfully completed when it is committed on the principal and the principal receives an acknowledgment from the mirror that the same transaction has been written and committed. This process guarantees transactional consistency between the principal and mirror; however, transaction commits and client performance may be hindered based on the network speed, mirror location, and available bandwidth between the principal and mirror server instances.

  The asynchronous transfer mechanism commits transactions to the principal database much faster because it does not require an acknowledgment from the mirror. This process does not guarantee transactional consistency between the principal and mirror.
- **Automatic and Manual Failover**—Database mirroring supports both an automatic and manual failover process between the principal and mirrored databases. The type of failover is dictated by the type of database mirroring configuration mode selected, whether a witness server is present, and the type of client used.
- **Transparent Client Redirect**—In the event of a failure, clients and applications automatically redirect from the principal database to the mirror database, resulting in minimal downtime. Be aware that automatic failover requires the latest SQL client based on the .NET and SQL Server Native Client (SNAC) providers/libraries.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 696

- **Database Mirroring Endpoint**—SQL Server 2005 uses endpoints to achieve secure server-to-server communication and authentication over the network. When you're configuring database mirroring, a dedicated endpoint is required exclusively for mirroring communications between both the principal and mirror database servers.

Database Mirroring Configuration/Operating Modes

Following are the database mirroring configuration and operating modes:

- **High Availability**—This database mirroring operating mode is also known as "Synchronous with a witness." It provides maximum availability for a database mirroring session by using a synchronous form of mirroring. This operating mode requires a witness and supports automatic detection and automatic failover in the event the principal database is unavailable. Client performance is affected by the network speed and distance between the principal and mirror servers.
- **High Protection**—High protection is also referred to as "Synchronous without a witness." Like the high-availability operating mode, high protection uses a synchronous form of mirroring but does not require a witness. It does not require a witness SQL Server instance because failover is manual. With this mode, just like the high-availability operating mode, principal performance is affected between the principal and mirror server based on network speed and distance.
- **High Performance**—High performance is the final operating mode and is also referred to as "asynchronous." High performance uses an asynchronous form of mirroring. In this situation, the principal server does not wait for confirmation that the transactions have been successfully committed to the mirror database instance. This increases performance because the network speed and distance are not factors. This solution does not require a witness. Therefore, there isn't automatic detection or automatic failover as in high-availability mode.

Summary of Database Mirroring Configuration Modes

Table 19.1 provides an easy-to-read summary of the database mirroring configuration modes, detection levels, and failover process; it also indicates whether a witness server is required.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 697

Table 19.1 **Database Mirroring Configuration Modes**

Database Mirroring Configuration Modes	Automatic Detection	Automatic or Manual Failover	Synchronous or Asynchronous	Requires Witness
High Availability	Yes	Automatic	Synchronous	Yes
High Protection	No	Manual	Synchronous	No
High Performance	No	Manual	Asynchronous	No

When you use high-availability and high-protection modes, principal performance is affected by network speed, distance, and available bandwidth. Performance is not affected when you use high performance. The mode you select ultimately determines how your organization wants to handle synchronization and failover processing.

SQL Server Database Mirroring Prerequisites

Configuring database mirroring is not as simple as clicking through a few screens of a SQL Server installation wizard. A number of prerequisite steps must be fulfilled before a database mirroring session can be configured and implemented. Following are the prerequisites:

- Register the principal, mirror, and witness SQL Server 2005 instances to ensure connectivity is present. The principal and mirror servers should be running the same edition of SQL Server 2005. Based on the features required, the Standard or Enterprise Edition can be used.
- The recovery model on the principal database must be set to Full.
- A full backup of the principal database is required.
- The mirror database needs to be initialized prior to implementing database mirroring by conducting a restore using the `NORECOVERY` option. All transaction logs taken after the full backup must also be restored.
- The database names for both the principal and mirror database must be identical.
- The server hosting the mirrored database requires adequate disk space.
- Database mirroring requires Service Pack 1 or higher.

When SQL Server 2005 Database Mirroring Is Desirable

Some of the key driving factors for implementing database mirroring with SQL Server 2005 include:

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 698

- There is a need to provide high-availability support for a specific database and not an entire instance of SQL Server.
- A seamless failover that does not affect client applications and end users is required.
- An automatic failover that does not require intervention from a database administrator is favorable.
- High availability for a database in another physical location other than the principal is required. Note that there isn't a distance limitation with mirroring.
- There is a need for high availability, and the organization does not have identical servers and shared storage, which is an expensive requirement for failover clustering.
- There is a need to fulfill the business continuity and disaster recovery requirements by placing and maintaining a redundant, up-to-date database in a different physical location than the principal.

There are many other reasons organizations may turn to database mirroring. The first step your organization should take is to identify the gaps between the current and desired states of your business and then determine if data mirroring fulfills your high-availability business goals.

Witness Server Placement

When an organization decides on using a witness server for high-availability and automatic failure, it is often challenged with where to place the server. If the database mirroring session is configured over a wide area network (WAN), the witness can be placed either in the site with the principal or the site with the mirror. It is recommended to place the witness server in the same site as the mirror server. The reason is that if a site failure occurs where the principal resides, the witness server will still be operational and can initiate the failover with ease. On the other hand, some organizations place the witness server in the same site as the principal server because the network may not be reliable between the two sites. For these organizations, placing the principal and witness together minimizes unnecessary failovers due to network glitches.

Finally, if your organization is using database mirroring as a high-availability alternative, the witness server should be configured on a dedicated server that is not the principal or mirror. Placement on a dedicated server in this situation protects against hardware failure. It is important to mention that even if

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 699

the witness is placed on a separate SQL Server instance, but the instance resides on the same server as the principal or mirror, you run into problems. If the physical hardware crashes, both instances fail and the witness cannot conduct an automatic failover, resulting in a longer downtime.

Combining Database Mirroring with Other SQL Server 2005 Technologies

Other SQL Server high-availability alternatives and technologies can be combined with database mirroring for maximum availability, reporting, business continuity, and disaster recovery. The following sections explain how database mirroring interacts with other SQL Server 2005 technologies.

Database Mirroring and Other High-Availability Alternatives

Database mirroring has its advantages and disadvantages, and it does not solve every high-availability requirement. This is why database mirroring can be combined with other SQL Server high-availability alternatives such as failover clustering, log shipping, and replication.

Database Mirroring and Failover Clustering

In many cases, database mirroring can be configured as a disaster recovery solution to a local SQL Server failover cluster instance by placing the principal database on the cluster and the hot standby mirror database in another physical location. If this combination is used, it is a best practice to use the high-protection or high-performance configuration mode because a cluster failover takes longer than the mirroring failover threshold. Therefore, if the high-availability configuration mode is being used, an automatic mirror failover takes place every time a cluster failover takes place between the two cluster nodes, making the cluster instance a mirrored database.

Database Mirroring and Log Shipping

One of the limitations of database mirroring compared to log shipping is that database mirroring can have only one mirrored server associated with each principal, whereas log shipping can have multiple standby servers. The two technologies can be combined if there is a need to ship the principal database logs to a remote location other than the place where the mirror resides. In addition, log shipping databases can be used for reporting, whereas mirror databases cannot unless a snapshot is used.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 700

Note

Log shipping needs to be reinitialized on the mirror SQL Server instance in the event of a failure or role change.

Database Mirroring and Replication

Finally, database mirroring can be used in conjunction with replication. The main focus is to provide availability for the publication database because the distribution and subscription databases are not supported with database mirroring. Because of the requirements and considerations, it is not a recommended practice to combine these two technologies; however, Microsoft includes a list of prerequisite tasks in SQL Server 2005 Books Online.

Database Mirroring and SQL Server 2005 Database Snapshots

Many organizations need to run reports against a production database for business purposes. To mitigate performance degradation and unnecessary locking due to sequential read and writes, it is a best practice to have a dedicated reporting server and not have reports run off the production database. Database mirroring offers this capability by allowing the mirror database to be used for reporting purposes. Unfortunately, the mirror database is in a constant recovering state, so it cannot be accessed directly. You can create a point-in-time database snapshot from the mirror database, which can be used for reporting.

Reporting against a mirrored database could be a great technology if offered by Microsoft because organizations would be able to create real-time reports off the mirror without the need to constantly create snapshots. Perhaps this could be a new feature if enough of us requested Microsoft to create it.

Note

For more information on creating database snapshots, see Chapter 17, "Backing Up and Restoring the SQL Server 2005 Environment" (online).

Administering a Database Mirroring Session

To administer a database mirroring session, first follow the steps to configure database mirroring. The following example simulates a database mirroring implementation that uses the high-availability configuration mode, including

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 701

a witness for CompanyABC's AdventureWorks production database located in San Francisco. The mirroring and witness partner is located in New York. The server names are shown in Table 19.2.

Table 19.2 **Roles and Server Names for Database Mirroring Example**

Role	SQL Server Instance	Location
Principal Server	SFC-SQL01\Principal	San Francisco
Mirror Server	NYC-SQL01\Mirror	New York
Witness Server	NYC-SQL01\Witness	New York

Configuring Database Mirroring Prerequisites

You must conduct the following tasks to configure database mirroring on the AdventureWorks database from San Francisco to New York:

1. From the principal server (SFC-SQL01\Principal), conduct a full backup of the AdventureWorks database by using Transact-SQL (TSQL) code or SQL Server Management Studio (SSMS).

Caution

Use independent files when creating the database and transaction log backups. Do *not* append both of these backups to the same file; otherwise, an erroneous error such as Error 1418 may occur when setting up the database mirroring session. Error 1418 typically represents network connectivity or issues when resolving server names in a mirroring session.

2. Conduct a transaction log backup of the AdventureWorks database by using TSQL code or SSMS.
3. Copy the backup files from the principal server (SFC-SQL01) to the mirror server (NYC-SQL01).

Tip

You need to create the AdventureWorks database on the mirror server if it does not already exist. To simplify the backup and restore process, it is a best practice to maintain the same file path for the database and transaction log files that the principal database is using. Otherwise, the `MOVE` command is required when you're conducting the restore.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 702

4. From the Mirror Server SQL Server instance (NYC-SQL01\Mirror), conduct a restore of the AdventureWorks database file and then the transaction log. Use the recovery state option `RESTORE WITH NORECOVERY` for both restores. Therefore, the database is not in an operational state ready to accept database mirroring transactions.

Note

For more information on backing up and restoring SQL Server with either SSMS or TSQL, see Chapter 17.

5. In SSMS, on the principal server, register the principal, mirror, and witness SQL Server instances to ensure successful connectivity and authentication.

Configuring Database Mirroring with High Availability and Automatic Failover

Now that you've configured the prerequisites, follow these steps to configure database mirroring with high safety and automatic failover:

1. From the principal server (SFC-SQL01), choose Start, All Programs, Microsoft SQL Server 2005, SQL Server Management Studio.
2. In Object Explorer, first connect to the Database Engine, expand the desired server (SFC-SQL01\Principal), and then expand the Database folder.
3. Right-click the AdventureWorks database, select Tasks, and then choose Mirror.
4. On the Database Properties screen, select the Configure Security button located on the Mirroring page.

Note

Because database mirroring requires the transaction logs for synchronization, you receive a warning message if the database recovery level is not set to Full. If this occurs, switch the recovery model to Full and restart the Database Mirroring Wizard.

5. On the Configure Database Mirroring Security Wizard Starting screen, select Next.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 703

6. Specify whether to include a witness server in the configuration by selecting the option Yes on the Include Witness Server screen. For this example, you use a witness server instance (NYC-SQL01\Witness) to operate database mirroring in synchronous mode with automatic failure.
7. In the Choose Servers to Configure screen, select the principal, mirror, and witness server instances, as illustrated in Figure 19.2. Click Next.

FIGURE 19.2
Configuring the database mirroring servers for security.

8. On the Principal Server Instance screen, specify the endpoint properties for the principal server instance, as shown in Figure 19.3. Click Next to continue.

Note

Typically, the default listener port for each endpoint is 5022. However, if the principal, mirror, or witness is configured on the same SQL Server instance, its endpoints must use different ports.

9. On the Mirror Server Instance screen, specify the mirror server instance and the endpoint properties for the mirrored server instance, as shown in Figure 19.4 Click Next to continue.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 704

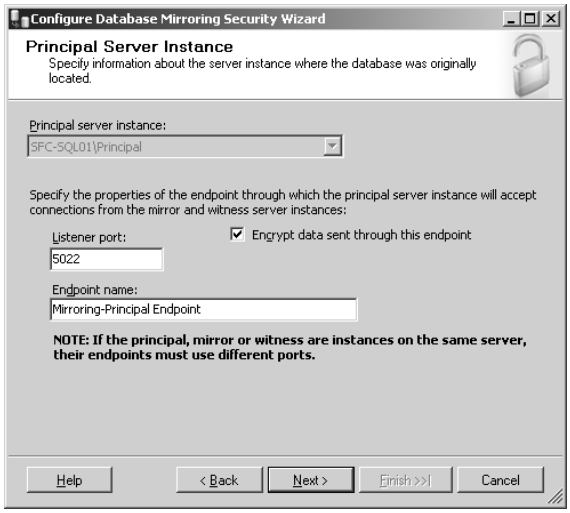

FIGURE 19.3
Entering the principal server instance information and settings.

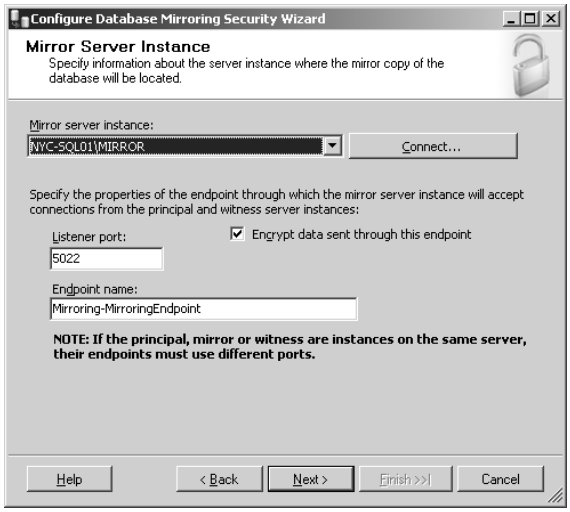

FIGURE 19.4
Entering the mirror server instance information and settings.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 705

10. On the Witness Server Instance screen, specify the witness server instance and the endpoint properties for the witness server instance, as shown in Figure 19.5. Click Next to continue.

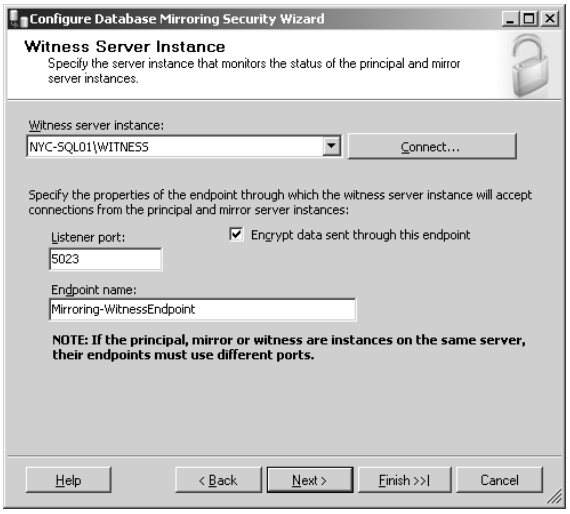

FIGURE 19.5
Entering the witness server instance information and settings.

11. On the Service Accounts screen, enter the service account information for each instance partaking in the database mirroring session. If the service accounts are the same for each instance, as in this example, leave the text boxes blank, as illustrated in Figure 19.6, and click Next to continue.

Note

If the service accounts entered are different and the accounts do not already exist in the specific SQL Server instance, the wizard automatically creates the accounts, grants appropriate permissions, and associates the account credentials to the endpoints.

12. On the Complete the Wizard screen, verify the configuration settings for each database mirroring instance, as shown in Figure 19.7, and then click Finish.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 706

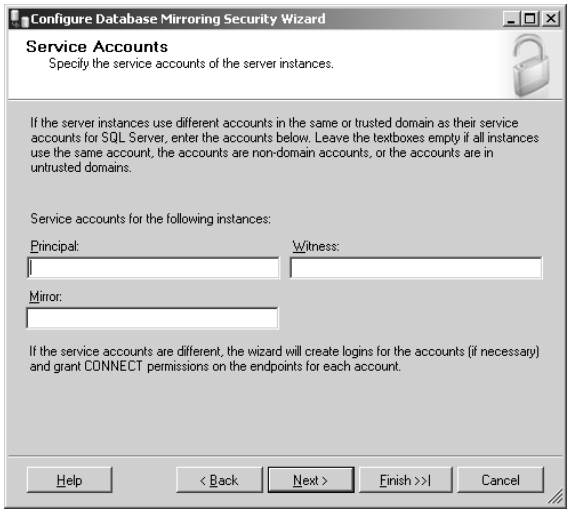

FIGURE 19.6
Specifying the database mirroring service accounts.

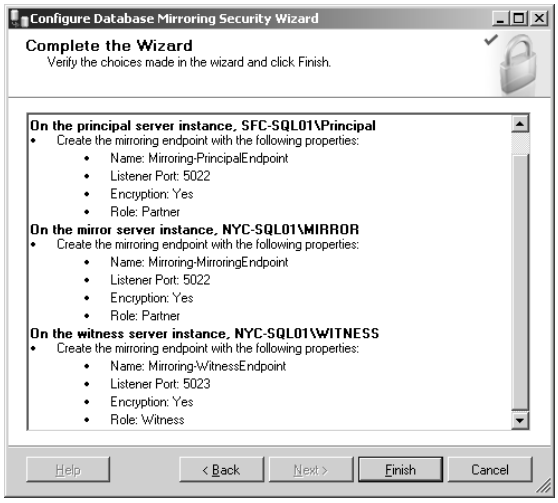

FIGURE 19.7
Verifying the database mirroring security settings.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 707

13. On the Configuring Endpoints screen, verify the status of each endpoint to ensure it was successfully created and click Close.
14. When this Endpoint Security Wizard is closed, you are prompted to either start the database mirroring session now by selecting Start Mirroring or start it later by selecting Start Mirroring on the Mirroring page of the Database Properties dialog box, as shown in Figure 19.8. For this example, click Start Mirroring.

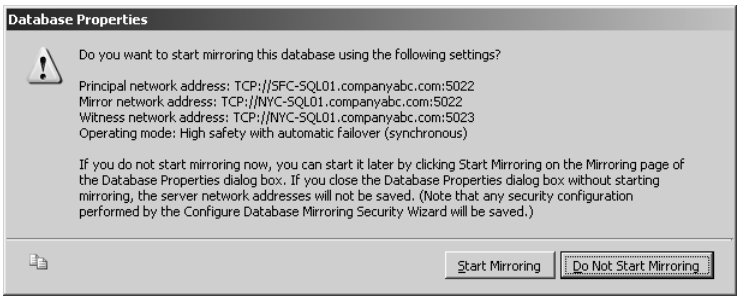

FIGURE 19.8
Starting the database mirroring.

Note

The mirrored database must be present on the mirrored server; otherwise, an error occurs, stating the mirrored database does not exist and must be created via a backup and restore prior to initializing the database mirroring session.

15. Verify the initial synchronization was successful by viewing the Status section located in the Database Properties page, as shown in Figure 19.9, and then click OK.

When databases are configured in a database mirroring session, a status message appears next to the database and includes the server role. For example, on the SFC-SQL01 server, the principal AdventureWorks database states `(Principal, Synchronized)`, and NYC-SQL01 states `(Mirror, Synchronized/Restoring...)`.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 708

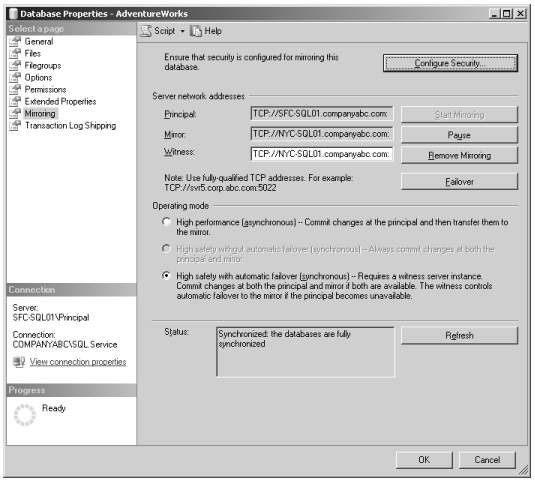

FIGURE 19.9
Viewing the database mirroring status in the Database Properties page.

Managing a Database Mirroring Session

The Database Mirroring page located on the Database Properties screen allows you to manage a database mirroring session. With this tool, you can pause, remove, or fail over a database mirroring session. In addition, it is possible to change the database mirroring operation mode—for example, from high performance (asynchronous) to high safety (synchronous). Finally, you can use this page to initiate manual failovers and status verification. Alternatively, any database mirroring tasks that can be conducted from the management console can also be scripted with Transact-SQL (TSQL).

Pausing and Resuming a Database Mirroring Session

Occasionally, you need to either pause or resume a database mirroring session for administrative purposes. You can pause and resume a mirroring session by using SSMS or TSQL.

Follow these steps to either pause or resume a database mirroring session with SSMS:

1. From the principal server (SFC-SQL01), choose Start, All Programs, Microsoft SQL Server 2005, SQL Server Management Studio.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 709

2. In Object Explorer, first connect to the Database Engine, expand the desired server (SFC-SQL01\Principal), and then expand the Database folder.
3. Right-click the AdventureWorks database, select Tasks, and then choose Mirror.
4. Click the Pause command button located in the Server Network Address section on the Mirroring tab of the Database Properties screen (refer to Figure 19.9).
5. Click the Resume command button to restart the database mirroring session. The Resume button is not displayed in Figure 19.10 because it only appears on the Mirroring tab once the Pause button has been clicked.

Alternatively, you can use the following sample TSQL syntax to pause and resume a database mirroring session:

Pausing database mirroring:

```
Use Master
ALTER DATABASE <database_name> SET PARTNER SUSPEND
GO
```

Resuming database mirroring:

```
Use Master
ALTER DATABASE <database_name> SET PARTNER RESUME
Go
```

Manually Failing Over a Database Mirroring Session

Follow these steps to swap the principal and mirror roles by manually failing over the database session from the principal server to the mirrored server:

1. From the principal server (SFC-SQL01), choose Start, All Programs, Microsoft SQL Server 2005, SQL Server Management Studio.
2. In Object Explorer, first connect to the Database Engine, expand the desired server (SFC-SQL01\Principal), and then expand the Database folder.
3. Right-click the AdventureWorks database, select Tasks, and then choose Mirror.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 710

4. Click the Failover command button located in the Server Network Addresses section on the Mirroring tab of the Database Properties screen.
5. Read the warning message and click Yes to finalize the role swap.

In SSMS, notice how the status messages have changed based on the role swap. On the SFC-SQL01 server, the AdventureWorks database states `(Mirror, Synchronized/In Recovery)`, whereas the NYC-SQL01 database instance states `(Principal, Synchronized)`.

The following sample TSQL syntax should be used to fail over a database mirroring session:

```
Use Master
ALTER DATABASE database_name SET PARTNER FAILOVER
Go
```

Changing the Database Mirroring Configuration/Operating Mode

In some situations, either you or your organization decides to change the operating mode. Reasons for changing the operating mode may include performance issues, the absence of a witness server, or even a need to manually control a failover as opposed to having SQL automatically conduct the role swap.

Follow these steps to change the database mirroring operating mode with SSMS:

1. From the principal server (SFC-SQL01), choose Start, All Programs, Microsoft SQL Server 2005, SQL Server Management Studio.
2. In Object Explorer, first connect to the Database Engine, expand the desired server (SFC-SQL01\Principal), and then expand the Database folder.
3. Right-click the AdventureWorks database, select Tasks, and then choose Mirror.
4. In the Operating Mode section, change the Operating Mode option to either High Performance, High Safety, or High Safety with Automatic Failover and click OK.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 711

You can use the following basic TSQL syntax to change the database mirroring operating mode:

Enable Transaction Safety:

```
Use Master
ALTER DATABASE <database> SET PARTNER SAFETY FULL
GO
```

Disable Transaction Safety:

```
Use Master
ALTER DATABASE <database> SET PARTNER SAFETY OFF
GO
```

Removing a Database Mirroring Session

Similar to the management steps you used previously, you can remove database mirroring sessions with either TSQL or SSMS.

Follow these steps to swap roles by manually failing over the database session from the principal server to the mirrored server:

1. From the principal server (SFC-SQL01), choose Start, All Programs, Microsoft SQL Server 2005, SQL Server Management Studio.
2. In Object Explorer, first connect to the Database Engine, expand the desired server (SFC-SQL01\Principal), and then expand the Database folder.
3. Right-click the AdventureWorks database, select Tasks, and then choose Mirror.
4. Click the Remove Mirroring command button located in the Server Network Addresses section on the Mirroring tab of the Database Properties screen.
5. Read the warning message and click Yes to remove mirroring from the AdventureWorks database.
6. In the Database Properties screen, click OK to finalize the procedures.

The following TSQL syntax can also be used to remove a database mirroring session:

```
Use Master
ALTER DATABASE <database_name> SET PARTNER OFF
Go
```

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 712

Managing Database Mirroring Client Connections and Redirect

In the event of a principal database failure, the principal database either fails over to the mirror manually or automatically. Therefore, all client connections need to be redirected from the principal server instance to the new mirror database instance. The latest ADO.NET or SQL Server clients have built-in redirect technologies that allow an application to automatically redirect its connection in the event of a database failure. Either you, as database administrator, or an application developer must specify the principal and failover SQL Server instance in the connection string to make this happen.

Follow these steps to configure automatic client redirect by using the native SQL Server client:

1. Choose Start, All Programs, Administrative Tools, Data Sources (ODBC).
2. On the ODBC Data Source Administrator screen, select System DNS.
3. Click Add to create a new System DSN connection to the principal and mirror SQL Server instance.
4. In the Create New Data Source screen, select SQL Native Client and then click Finish.
5. In the Create a New Data Source to SQL Server screen, enter the name, description, and the principal database server instance, as illustrated in Figure 19.10. For this example, use the principal SQL Server instance SFC-SQL01\Principal. Click Next.

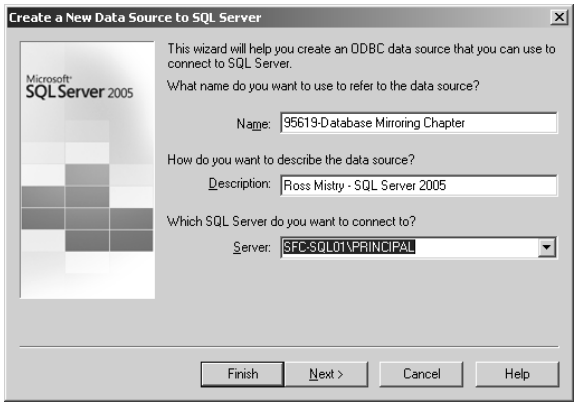

FIGURE 19.10
Creating a new SQL Server native client data source.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 713

6. Select the SQL Server authentication mode for the SQL Server connection and click Next.
7. Select the default database to connect to and enter the name of the mirror server, as shown in Figure 19.11. For this example, select AdventureWorks database and NYC-SQL01\Mirror for the mirror server instance. Click Next.

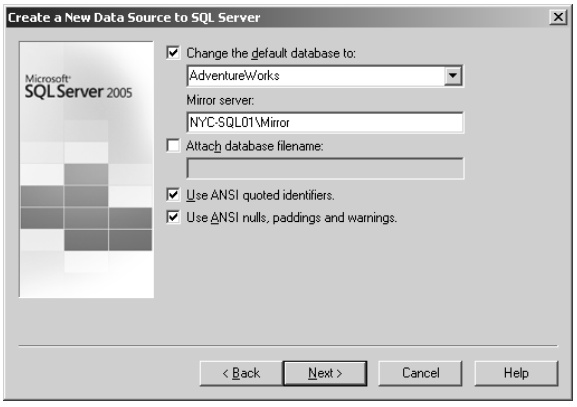

FIGURE 19.11
Specifying the mirror database settings.

8. Click Finish and then click Test Data Source to finalize the connection settings.

The new connection can be leveraged with a front-end SQL Server client such as Access, Visual Studio .NET, or Reporting Services. Use the newly created connection and display data from the AdventureWorks database such as the Employee table. When a connection is established and the data is presented, fail over the database mirroring session. The application should still be able to display the Employee table because it automatically redirects to the AdventureWorks database residing on the mirror instance of SQL Server.

Monitoring and Troubleshooting a Database Mirroring Session

After you have configured database mirroring, you should turn your attention to understanding the following tools available for monitoring and managing the mirroring session:

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 714

- Database Mirroring Monitoring tool
- System Performance
- System Catalogs
- Operations Manager 2007

Using the Database Mirroring Monitoring Tool to Manage Database Mirroring

The Database Mirroring Monitoring tool is included with SSMS and should be used to monitor databases configured in a mirroring session. The tool can be launched by right-clicking the database partaking in a database mirroring session and then selecting Tasks, Launch Database Mirroring Monitor. You can use the tool to identify the status of the database mirroring session, identify the role of each partner, and determine whether the mirroring session is behind schedule and estimate the time it will take to catch up.

Use the following procedure to monitor the state of the database mirroring session configured in the earlier examples:

1. From the principal server (SFC-SQL01), choose Start, All Programs, Microsoft SQL Server 2005, SQL Server Management Studio.
2. In Object Explorer, first connect to the Database Engine, expand the desired server (SFC-SQL01\Principal), and then expand the Database folder.
3. Right-click the AdventureWorks database, select Tasks, and then choose Launch Database Mirroring Monitor.
4. To register a mirrored database, either click the Register Mirrored Database hyperlink in the right pane or select Action, Register Mirrored Database from the Tool menu.
5. In the Register Mirrored Database screen, select the server instance by clicking the Connect command button.
6. In the Connect to Server dialog box, select the Mirrored Database SQL Server Instance (NYC-SQL01\Mirror) and click OK.
7. In the Register Mirrored Database screen, click the Register check box next to the database to register the mirrored database instance, as shown in Figure 19.12, and then click OK.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 715

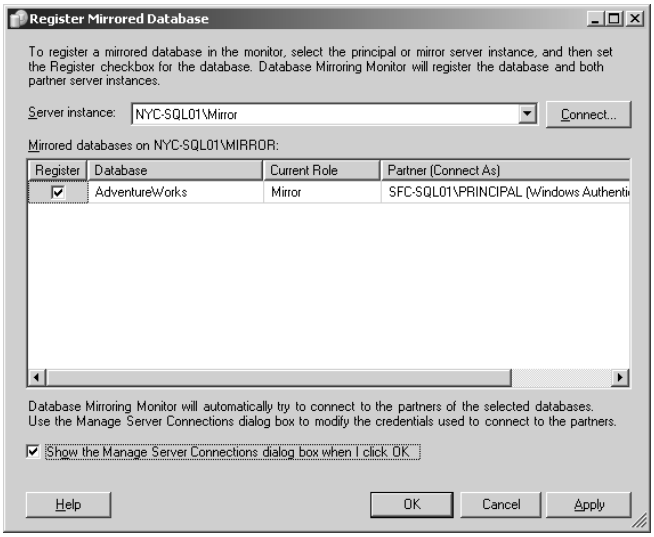

FIGURE 19.12
Registering a mirrored database for database mirroring monitoring.

8. The Database Mirroring Monitoring tool automatically connects to both the principal and mirror instances partaking in the database mirroring session. In the Manage Server Instance Connections, edit the credentials if necessary or click OK.

The Database Mirroring Monitoring Tool Status Tab

The Status tab includes a plethora of database mirroring status information for both the principal and mirror databases. The status information located on the Status tab is broken into four sections: Status, Principal Log, Mirror Log, and General Information.

The Status section indicates the server instance, current role, and mirrored state, and it validates that the witness is operational. The final command in the status window provides a history log file, as shown in Figure 19.13.

The Principal Log section includes metrics on the following:

- Unsent Log Information in KB
- Oldest Unsent Transaction
- Time to Send Log Estimate

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 716

- Current Send Rate in KB per Second
- Current Rate of New Transactions

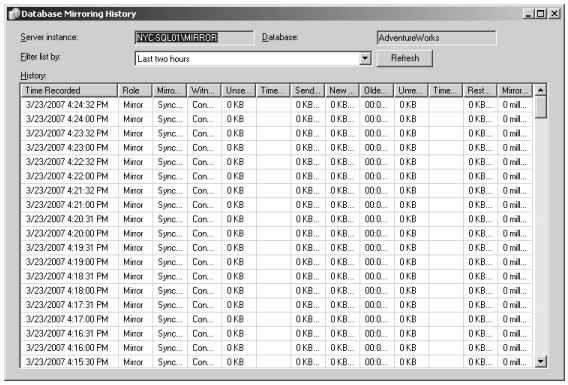

FIGURE 19.13
Displaying database mirroring history.

The General section located at the bottom of the Status tab page includes additional status for troubleshooting and diagnostics:

- Mirror Commit Overhead in Milliseconds
- Time Estimates to Send and Restore All Current Logs
- Witness Address
- Operation Mode

The Database Mirroring Monitoring Tool Warnings Tab

The Warnings tab allows you to set database mirroring warning thresholds for the principal and mirror SQL Server instances. The four warnings included with this tool are

- Warn If the Unsent Log Exceeds the Threshold
- Warn If the Un-Restored Log Exceeds the Threshold
- Warn If the Age of the Oldest Unsent Transaction Exceeds the Threshold
- Warn If the Mirror Commit Overhead Exceeds the Threshold

The Set Warning Thresholds screen should be used to enable/disable warning per instance and set thresholds.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 717

Monitoring Database Mirroring Performance

The Database Mirroring Monitoring tool is a great starting point for managing and analyzing a database mirroring session. When additional metrics are needed for analysis or when troubleshooting or creating a performance baseline, you can use the Performance Monitor performance monitoring tool included with Windows Server 2003. To launch the tool, choose Start, All Programs, Administrative Tools and Performance.

Following are the specific counters included with the SQL Server Database Mirroring Performance Object:

- Bytes Received/sec
- Bytes Sent/sec
- Log Bytes Received/sec
- Log Bytes Sent/sec
- Log Send Queue KB
- Pages Sent/sec
- Receives/sec
- Redo Bytes/sec
- Redo Queue KB
- Send/Receive Ack Time
- Sends/sec
- Transaction Delay

Collecting and analyzing the preceding metrics assists organizations with planning their database mirroring solution. Before database mirroring is implemented in production, it is a best practice to simulate mirroring in a prototype test lab and analyze the metrics collected. If possible, a bandwidth simulator tool should also be used to mimic the production network speed. This allows an organization to fully understand the database mirroring and bandwidth requirements when setting up database mirroring in production over a private network. When analyzing bandwidth requirements, your organization should also assess the current bandwidth utilization. Therefore, if the link is already fully saturated, more bandwidth may be necessary to support the mirroring solution. Alternatively, many organizations purchase dedicated network lines tailored specifically for database mirroring replication.

Using the System Catalogs to Monitor Database Mirroring

The catalog view included with SQL Server is another great source of information when monitoring status and performance.

The following catalog views should be used:

- `Sys.database_mirroring`
- `Sys.database_mirroring_witness`
- `Sys.database_mirroring_endpoints`

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 718

- `Sys.tcp_endpoints`
- `Sys.Server_principals`
- `Sys.Server_recovery_status`

The catalog view provides database mirroring metadata for a session, witness, endpoint, principal, and recovery status.

Monitoring Database Mirroring with Operations Manager 2007

Another great tool to proactively monitor database mirroring, including the health of the principal, mirror, and witness SQL Server instances, is Microsoft Operations Manager 2007. Operations Manager 2007 includes a dedicated Microsoft Management Pack tailored toward SQL Server. It includes a subcomponent that focuses on database mirroring.

> **Note**
>
> For more information on proactively monitoring a database mirroring session with Operations Manager 2007, refer to Chapter 21, "Monitoring SQL Server 2005" (online).

Summary

Database mirroring is a SQL Server 2005 high-availability alternative that can be used for maintaining a redundant copy of the principal database on a standby server for increased availability and disaster recovery purposes.

How well database mirroring performs is closely associated with the type of application, transaction safety level, and network performance between the principal and mirror servers. Understanding the application behavior in terms of the log generation rate, number of concurrent connections, and size of transactions is important in achieving the best performance.

In addition, the network plays a very important role in a database mirroring environment. When used with a high-bandwidth and low-latency network, database mirroring can provide a reliable high-availability solution against planned and unplanned downtime. With data centers in different geographical locations, database mirroring can provide the foundation for a solid, inexpensive disaster recovery solution.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 719

Best Practices

The following are the best practices for this chapter:

- Database mirroring using the high-availability configuration mode is a practical alternative when the principal and mirror server reside in the same physical location. The reason is that most organizations' production environments are running fast networks without network latency.
- Database mirroring using the high-performance configuration mode is a practical alternative when the principal and mirror server reside in different physical locations. The reason is that production performance is typically of higher importance than automatic failover and availability in these situations.
- Leverage database mirroring to reduce planned downtime, increase availability for mission-critical databases, and satisfy disaster recovery requirements.
- To increase performance, implement and leverage a dedicated high-bandwidth network for synchronization communications between the principal and mirror database servers when possible.
- In the event of a failure, the mirror server needs to maintain the same workload as the principal. Both servers should be of similar class and have the same number of processors and same amount of memory and storage. Unlike in failover clustering, the hardware does not have to be an exact match, but the mirror needs to support the same load as the principal.
- Use failover clustering over database mirroring if there is a need to provide high availability on the whole SQL Server instance. This includes the master, model, msdb, and tempdb databases because these system databases cannot partake in a database mirroring session.
- To reduce the number of unforeseen issues with database mirroring, use the same edition of Windows and SQL Server for both the principal and mirror server. In addition, the service packs, hotfixes, drive letter layout, collation settings, and SQL Server configuration settings should be identical. Although this is not a requirement, it is a best practice.
- To reduce complications and troubleshooting, use a single mirror SQL Server instance if a principal instance is composed of multiple databases belonging to one application that needs to be mirrored.

25_0672329565_ch19.qxd 9/7/07 8:37 AM Page 720

- When using database mirroring, create items such as logins, scheduled jobs, and extended stored procedures that are identical on the mirrored database and instance.
- When configuring database mirroring, do *not* forget to initialize the mirror database by restoring the full backup and the last transaction log with the `NORECOVERY` option.
- If you configured the database mirroring session to use high-availability or high-protection mode and delays are experienced with client applications, switch to high-performance mode.