When you set up database mirroring, the experience can range from the incredibly simple to the overly complex. SQL Server offers a database mirroring wizard to guide you through the simple setups. The wizard is quick and easy and really is the way to go when you just need to get the basics up and running.

If you have multiple databases to mirror, the wizard can be cumbersome and slow you down. Likewise, if you want to use the more advanced features of database mirroring, you will have to set up mirroring with T-SQL. Once you get accustomed to setting up mirroring with T-SQL, this method will be much faster for you. We will cover both methods of database mirroring setup in this chapter.

---

Note For the screenshots and sample code included in this chapter, we are using a database named MirrorTest. You should replace MirrorTest with your own database’s name when you use the provided code.

---

Before You Begin

You need to perform certain steps before setting up mirroring. You need to perform these preparatory steps regardless of the strategy you are planning to use for setting up mirroring. The database mirroring wizard does not prepare the database for mirroring.

You can split the preparation into two separate parts. Perform the first part in advance of the database mirroring setup. For the most part, it really doesn’t matter how far in advance you perform the steps in part one; these are just prerequisites for database mirroring and your application. You will need to perform the second part immediately prior to setting up mirroring because it interrupts the routine maintenance of your database such as backups.

Part One: Advanced Preparation

You should configure as much as you can prior to actually setting up database mirroring. This is especially helpful if you are setting up database mirroring during an outage window. The more you can do in advance, the more time you have to focus on the tasks you need to perform immediately prior to setting up database mirroring. Follow these steps to prepare your environment for database mirroring:

1. Set the recovery model of the database to the Full Recovery model if not already set. You can make this change in SQL Server Management Studio (SSMS) on the Options tab of the Database Properties dialog. You can also manually make the change with the following T-SQL code:

   \[
   \text{Alter Database [MirrorTest] Set Recovery Full;}
   \]
2. Set the compatibility level of your database to level 90 or higher. Level 90 is SQL Server 2005 compatibility, and level 100 is SQL Server 2008 compatibility. You can set the compatibility level of your database in SSMS on the Options tab of the Database Properties dialog, or you can manually make the change with T-SQL. You can use the following commands to change your database's compatibility level.
For SQL Server 2005:
```sql
Exec sp_dbcmptlevel @dbname = 'MirrorTest', @new_cmptlevel = 90
```
For SQL Server 2008:
```sql
Alter Database [MirrorTest] Set Compatibility_Level = 100;
```
3. Copy all server logins from the principal server instance to the mirror server instance. There are several ways to do this.
   • SQL Server Integration Services (SSIS) has a transfer logins task you can use to copy logins to the other server.
   • You can use the Generate Scripts wizard by right-clicking on your database, highlighting Tasks, and clicking on Generate Scripts. Select the option to script logins and select all Users as the objects to script. Output the script to a new query window and run the login create statements on the mirror server instance.
   • Query the system security views to create your own T-SQL code to create the logins. We will show you how to script this approach in Chapter 6 so you can set up a maintenance process to keep the logins in sync.
4. Set up any external resources that you need on the mirror server such as SSIS packages, routine maintenance jobs, file shares, and so on.

Depending on how far in advance you perform the preliminary setup, you may need to validate that what you already set up is current. Even if there was a code freeze on the servers, you may need to add or delete server logins or change the permissions for a login. If logins or external resources have been added or changed on the principal server, you need to make sure the changes are applied to the mirror server as well.

Part Two: Pre-setup Configuration

The purpose of part two is to move a current copy of the principal database to the mirror server. Your data needs to be as current as possible so database mirroring does not have to try to apply too many transactions to the mirror database while they are trying to synchronize. Perform the following steps immediately before you begin configuring database mirroring:

1. Disable any jobs on the server that perform full backups or log backups. If a full or log backup is created in between the time that you create your full backup (in the next step) and the time that you create your log backup (in Step 4), you will not be able to restore the log backup that you created over the full backup that you created.
2. Create a full backup of your database and copy it to the mirror server instance.
3. Restore the full backup on the mirror instance using the NORECOVERY option. If you are using the Restore Database User Interface, you can specify NORECOVERY on the Options tab under Recovery State. Select the second recovery state (Restore With NoRecovery).
4. Create a log backup of your database and copy it to the mirror server instance.
5. Restore the log backup on the mirror instance using the NORECOVERY flag. If you are using the Restore Database User Interface, you can specify NORECOVERY on the Options tab under Recovery State. Select the second recovery state (Restore With NoRecovery).