SearchSQLServer.com

Database mirroring setup in SQL Server 2005

One of many new features in SQL Server 2005, database mirroring allows you to automatically mirror database contents from one SQL Server database to another. It also offers failover capabilities, which may be made automatic depending on how you configure mirroring. The mirrored copy is a standby copy that can not be accessed directly. It is used only for failover situations. With SQL Server 2005 Enterprise Edition you can also use database snapshots with the mirror, but we will cover that another time.

The following table of contents will help you navigate this tip.

TABLE OF CONTENTS

How it works

You need a minimum of two different SQL Server instances for database mirroring to work. The primary instance is the "principal." The secondary instance is the "mirror." The principal is your live database and the mirror is your standby copy of the database. As transactions are written to your principal database they are sent to your mirrored database and written there as well.

In addition to the principal and the mirror, another optional component called the "witness" can be introduced. The witness is a third instance of SQL Server 2005 that acts as an intermediary between the principal and the mirror to determine when to failover. This option is only used when you want to run an automatic failover. It creates the ability to have a 2-to-1 vote that says one of my components is not available and therefore I am going to failover. The witness server is only needed when you want to implement automated failover.

Follow this link for a SQL Server 2005 database mirroring primer.

How to implement it

Database mirroring offers three modes of implementation. The choice you select depends on how you want to handle failover processing.

- **High availability:**
  - This option allows you to synchronize transaction writes on both servers and you to automate failover.
  - Database instances needed: principal, mirror and witness

- **High protection:**
  - This option allows you to synchronize transaction writes on both servers, but failover is manual. - Database instances needed: principal and mirror

- **High performance:** - This option does not care if writes are synchronized on both servers and, therefore, offers some performance gains. When using this option you assume that everything is going to complete successfully on the mirror and failover is a manual process. - Database instances needed: principal and mirror

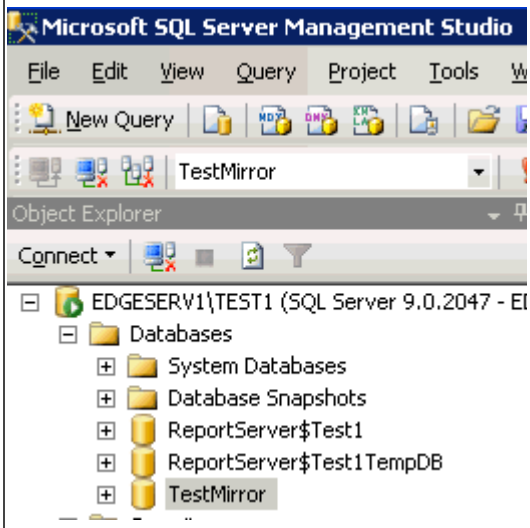
How to set up

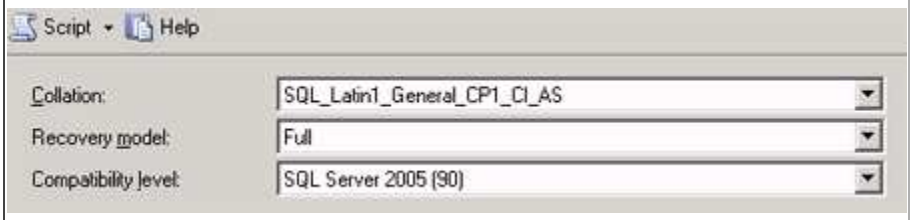
Once you have selected instances and the mode to use, a few other requirements must be met. You must have SQL Server 2005 Standard, Enterprise or Developer editions plus SQL Server 2005 Service Pack 1 to run the principal and the mirror. Prior to SP1, database mirroring could be set up using a trace flag, but it was not supported by Microsoft. For the witness, other versions of SQL Server 2005 can be used.

Database mirroring setup via SQL Server Management Studio

As with most things in SQL Server, you have the option to use GUI tools and an option to use T-SQL commands. For this tip I will focus on setup using SQL Server Management Studio.

To get started select the database and instances for the principal, mirror and the witness, if you are going to use one. Again these only need to be unique instances, so they can all be on the same physical server. For testing and development this makes sense, but for your production environment the whole idea of mirroring is to use physically different servers; if there is a problem with your primary server you can fail over to a secondary server. Let's begin the setup:

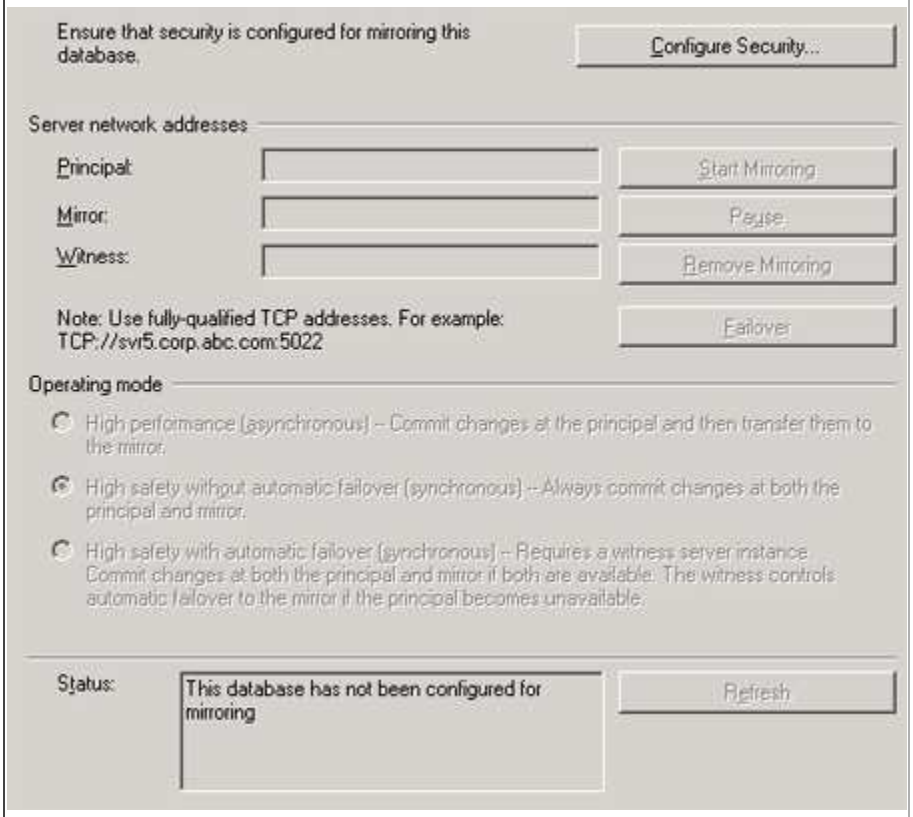
Database Mirroring Quick Setup	
Steps	**Screen shot/command**
Select the database you want to mirror using SQL Server Management Studio.	

Make sure the database is in the **Full Recovery mode**. To do this, right click the database name, select Properties and then select Options.	
Run a **full backup** of your database. This can be done through the GUI or using a T-SQL command.	BACKUP TestMirror TO DISK='C:\Backup\ TestMirror_FULL.BAK
Run a **restore** of this backup on your mirror. This can be done through the GUI or using a T-SQL command. The database restore must use the NO RECOVERY option, so the database stays in a loading state. Also the database name on the mirror must be the exact same name as the principal.	RESTORE TestMirror FROM DISK='C:\Backup\ TestMirror_FULL.BAK' WITH NORECOVERY You will probably need to use the WITH MOVE option to specify the new drive and directory path for the data and log files.

Select your **principal database**.

Right click on the database name and select Properties and the screen to the right will appear.

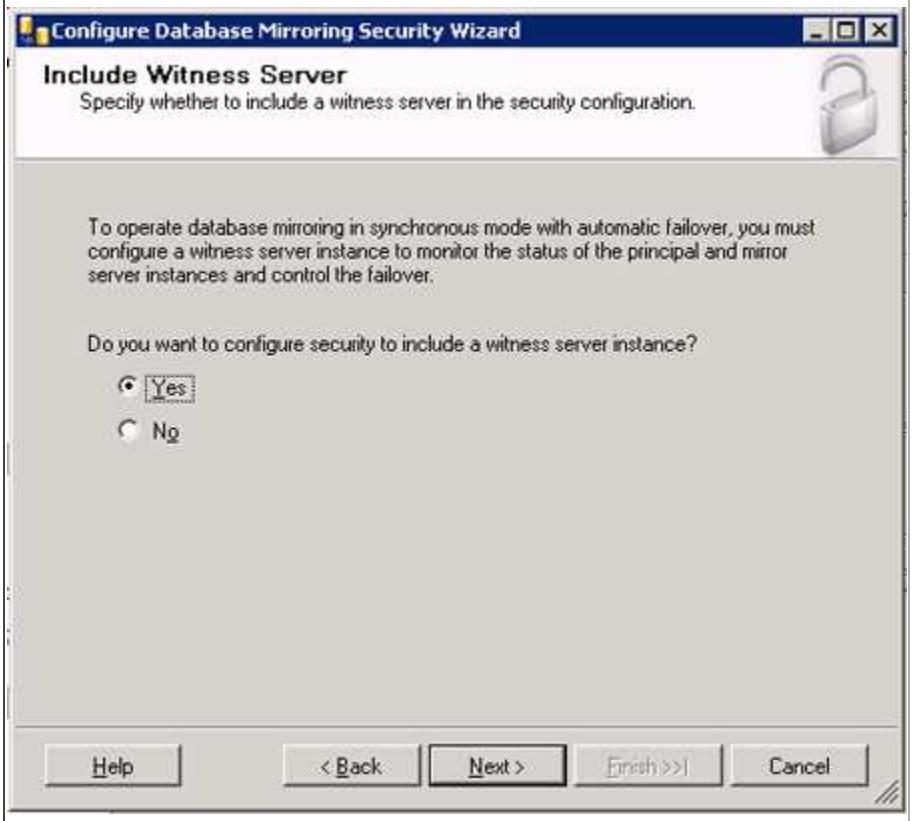
Click on the "Configure Security. . ." button.



Click Next to get started.

Select whether you want to use a **witness server** or not.

Click Next.

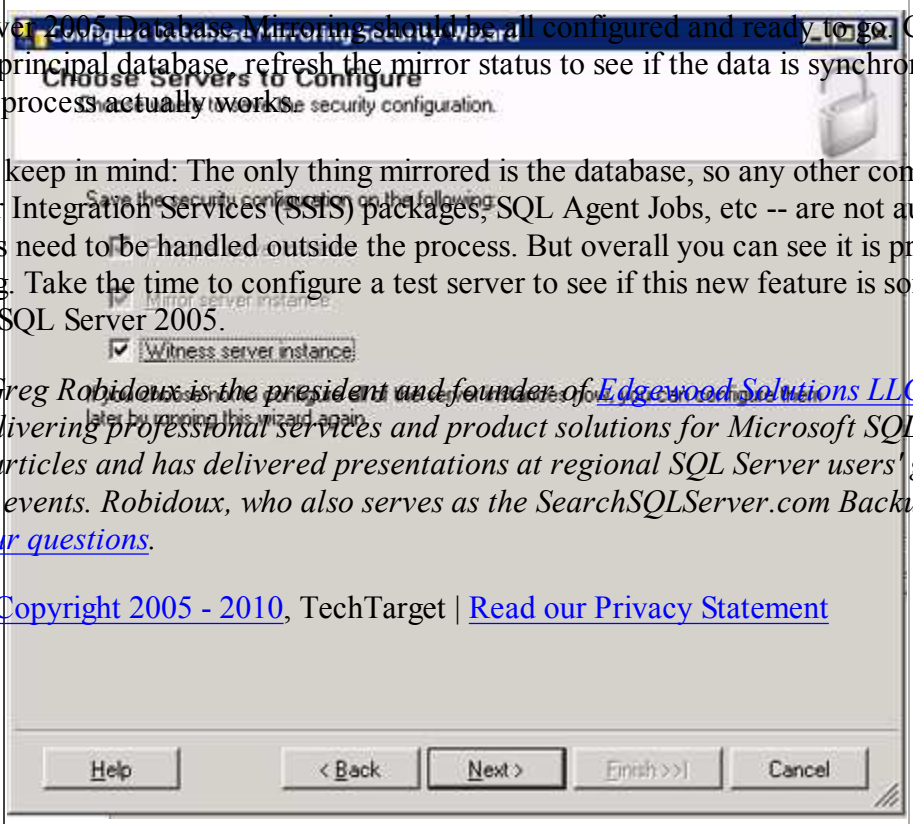


At this time SQL Server 2005 Database Mirroring should be all configured and ready to go. Create some transactions on your principal database, refresh the mirror status to see if the data is synchronized and then fail over to see if the process actually works.

A couple of things to keep in mind: The only thing mirrored is the database, so any other component -- such as logins, SQL Server Integration Services (SSIS) packages, SQL Agent Jobs, etc -- are not automatically mirrored. These items need to be handled outside the process. But overall you can see it is pretty easy to set up database mirroring. Take the time to configure a test server to see if this new feature is something you can take advantage of in SQL Server 2005.

Again select whether you want to use a witness server.

Click Next.



About the author: *Greg Robidoux is the president and founder of Edgewood Solutions LLC, a technology services company delivering professional services and product solutions for Microsoft SQL Server. He has authored numerous articles and has delivered presentations at regional SQL Server users' groups and national SQL Server events. Robidoux, who also serves as the SearchSQLServer.com Backup and Recovery expert, welcomes your questions.*