

Build Database Mirroring

Author	Avi Niv	Create Date	12-DEC-2013
Revised By		Revision Date	
– DRAFT –			

Guidelines

- Check list items are marked with a checkbox at the beginning.
- Action items are numbered in a sequential order, continue to the next action item only if your current action item completed successfully.
- All disk operations are performed on the "C:\Temp\Mirroring" directory.

Terms

- Mirroring – the process of moving the database level changes from in-use database to another database.
- Quorum – a connection of at least 2 servers configured in the mirroring.
- Server – a SQL Server instance.
- Principal – the source (in-use) database. Can be referred also as the principal server. Named as ServerA.
- Mirror – the target (stand-by) database. Can be referred also as the mirror server. Named as ServerB.
- Partner – either the principal or mirror. Named as ServerC.
- Witness – a server used to watch the partners status.
- Failover – a process of switching the role of the partners.
- End-point – a gateway in the instance layer, used by the mirroring to transfer data.

Considerations

- A server can have only one DATABASE_MIRRORING end point.
- A server can have multiple mirrored databases, each of which using the same witness.
- All connections to the end-points are encrypted by default.
- When using certificate for authentication, outbound must be configured before inbound.
- Prefer using windows (active directory) accounts.
- The default mirroring time-out period is 10 seconds, and should be considered as the minimum.
- Any relevant job referring the database should be copied to the mirror server.
- It is recommended to create a job enabling/disabling methodology for cases of failover.

Prerequisites

- ☐ Make sure you fully understand the concept, deployment and monitoring of Database Mirroring.
- ☐ Both principal and mirror databases must be running on the same version and edition of SQL Server.
- ☐ The partners' server version must be standard or above.
- ☐ The witness' server version can be express or above.
- ☐ Principal database must be in full recovery model.
- ☐ Both principal and mirror databases must have the same name.
- ☐ The principal database is not part of an AlwaysOn Availability Group.
- ☐ Do not reconfigure an in-use database mirroring endpoint.

- ☐ Make sure all end-point ports on servers are not in use.
- ☐ In case the servers are connected via a firewall, verify there is a rule for the ports used for mirroring.
- ☐ Preferably, the path of the mirror database should be identical to the path of the principal database.
- ☐ Transfer SQL Server logins from principal to mirror using sp_help_revlogin.
- ☐ Recreate each relevant job in the principal server to the mirror server.
- ☐ Verify sufficient disk space for the mirror database and backup files.
- ☐ A sysadmin permissions is required on all related servers (principal, mirror and witness).

1. Create Network Objects – Outbound

In this section, you will create a certificate to be used in the creation of a new database mirroring end-point. The certificate will later be used to connect to the other servers. This process should be executed on all servers in the quorum.

If one of the partner role servers is designated to be also the witness server, define the ALL role (instead of PARTNER).

Reminder: only one mirroring end point can exist on each server.

You can use the following queries to check the current status of your progress:

- Query for the master key in the symmetric keys DMV:

```
SELECT [name], [algorithm_desc], [create_date], [modify_date]
FROM [master].sys.symmetric_keys
WHERE [name] = '##MS_DatabaseMasterKey##';
```

- Query for a certificate, filtering by the naming convention of "cert_<ServerName>":

```
SELECT [name], [pvt_key_encryption_type_desc], [subject], [expiry_date]
FROM [master].sys.certificates
WHERE [name] LIKE '%cert[_]%%';
```

- Query for database mirroring end-points:

```
SELECT [name], [role_desc], [state_desc], [connection_auth_desc], [encryption_algorithm_desc]
FROM [master].sys.database_mirroring_endpoints;
```

1.1 On ServerA (will initially serve as Principal):

- ☐ -- Create Master Key


```
USE [master];
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'PWD1!';
GO
BACKUP MASTER KEY
  TO FILE = 'C:\Temp\Mirroring\master_key'
  ENCRYPTION BY PASSWORD = N'PWD2@';
GO
```
- ☐ -- Create Certificate


```
USE [master];
GO
CREATE CERTIFICATE [cert_ServerA]
  WITH SUBJECT = 'Certificate for Database Mirroring (ServerA)', EXPIRY_DATE = '2050-01-01';
GO
BACKUP CERTIFICATE [cert_ServerA] TO FILE = 'C:\Temp\Mirroring\cert_ServerA.cer';
GO
```

```
-- Create Certificate-Based Mirroring End-Point
CREATE ENDPOINT Endpoint_Mirroring
STATE = STARTED
AS TCP ( LISTENER_PORT=7024 , LISTENER_IP = ALL )
FOR DATABASE_MIRRORING (
    AUTHENTICATION = CERTIFICATE [cert_ServerA]
    , ENCRYPTION = REQUIRED ALGORITHM AES
    , ROLE = PARTNER
);
GO
```

1.2 On ServerB (will initially serve as **Mirror**):

```
-- Create Master Key
USE [master];
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'PWD1!';
GO
BACKUP MASTER KEY
TO FILE = 'C:\Temp\Mirroring\master_key'
ENCRYPTION BY PASSWORD = N'PWD2@';
GO

-- Create Certificate
USE [master];
GO
CREATE CERTIFICATE [cert_ServerB]
WITH SUBJECT = 'Certificate for Database Mirroring (ServerB)', EXPIRY_DATE = '2050-01-01';
GO
BACKUP CERTIFICATE [cert_ServerB] TO FILE = 'C:\Temp\Mirroring\cert_ServerB.cer';
GO

-- Create Certificate-Based Mirroring End-Point
CREATE ENDPOINT Endpoint_Mirroring
STATE = STARTED
AS TCP ( LISTENER_PORT=7024 , LISTENER_IP = ALL )
FOR DATABASE_MIRRORING (
    AUTHENTICATION = CERTIFICATE [cert_ServerB]
    , ENCRYPTION = REQUIRED ALGORITHM AES
    , ROLE = PARTNER
);
GO
```

1.3 On ServerC (will initially serve as **Witness**):

```
-- Create Master Key
USE [master];
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'PWD1!';
GO
BACKUP MASTER KEY
TO FILE = 'C:\Temp\Mirroring\master_key'
ENCRYPTION BY PASSWORD = N'PWD2@';
GO

-- Create Certificate
USE [master];
GO
CREATE CERTIFICATE [cert_ServerC]
WITH SUBJECT = 'Certificate for Database Mirroring (ServerC)', EXPIRY_DATE = '2050-01-01';
GO
BACKUP CERTIFICATE [cert_ServerC] TO FILE = 'C:\Temp\Mirroring\cert_ServerC.cer';
GO

-- Create Certificate-Based Mirroring End-Point
CREATE ENDPOINT Endpoint_Mirroring
STATE = STARTED
AS TCP ( LISTENER_PORT=7024 , LISTENER_IP = ALL )
FOR DATABASE_MIRRORING (
    AUTHENTICATION = CERTIFICATE [cert_ServerC]
    , ENCRYPTION = REQUIRED ALGORITHM AES
    , ROLE = WITNESS
);
GO
```

1.4 Copy the certificates to the other servers, so any server will have all 3 certificates.

2. Create Network Objects – Inbound

In this section, you will create logins, users and certificates to be used by the other servers to access this server. This process should be executed on all servers in the quorum.

Reminder: the other servers' certificate backup file should be accessible to the current server.

Note: the users will be created in the [master] database.

You can use the following queries to check the current status of your progress:

- Query for a certificate, filtering by the naming convention of "cert_<ServerName>":

```
SELECT [name], [pvt_key_encryption_type_desc], [subject], [expiry_date]
FROM [master].sys.certificates
WHERE [name] LIKE '%cert[_]%%';
```

- Query for a login, filtering by the naming convention of "MirroringLogin_<ServerName>":

```
SELECT [name], [type_desc], [is_disabled], [default_database_name]
FROM [master].sys.server_principals
WHERE [name] LIKE 'MirroringLogin[_]%%';
GO
```

- Query for a user, filtering by the naming convention of "MirroringUser_<ServerName>":

```
SELECT [name], [type_desc], [authentication_type_desc]
FROM [master].sys.database_principals
WHERE [name] LIKE 'MirroringUser[_]%%';
GO
```

2.1 On ServerA (connection from ServerB):

- ☐ -- Create Login for Mirror (ServerB)
USE [master];
GO
CREATE LOGIN [MirroringLogin_ServerB] WITH PASSWORD = 'PWD1!';
GO
CREATE USER [MirroringUser_ServerB] FOR LOGIN [MirroringLogin_ServerB];
GO
- ☐ -- Create Certificate for Mirror (ServerB)
CREATE CERTIFICATE [cert_ServerB]
AUTHORIZATION [MirroringUser_ServerB]
FROM FILE = 'C:\Temp\Mirroring\cert_ServerB.cer'
GO
- ☐ -- Grant Connection to the End-Point
GRANT CONNECT ON ENDPOINT::Endpoint_Mirroring TO [MirroringLogin_ServerB];
GO

2.2 On ServerA (connection from ServerC):

- ☐ -- Create Login for Witness (ServerC)
USE [master];
GO
CREATE LOGIN [MirroringLogin_ServerC] WITH PASSWORD = 'PWD1!';
GO
CREATE USER [MirroringUser_ServerC] FOR LOGIN [MirroringLogin_ServerC];
GO
- ☐ -- Create Certificate for Witness (ServerC)
CREATE CERTIFICATE [cert_ServerC]
AUTHORIZATION [MirroringUser_ServerC]
FROM FILE = 'C:\Temp\Mirroring\cert_ServerC.cer'
GO
- ☐ -- Grant Connection to the End-Point
GRANT CONNECT ON ENDPOINT::Endpoint_Mirroring TO [MirroringLogin_ServerC];
GO

2.3 On ServerB (connection from ServerA):

- ☐ -- Create Login for Mirror (ServerA)
USE [master];
GO
CREATE LOGIN [MirroringLogin_ServerA] WITH PASSWORD = 'PWD1!';
GO
CREATE USER [MirroringUser_ServerA] FOR LOGIN [MirroringLogin_ServerA];
GO
- ☐ -- Create Certificate for Mirror (ServerA)
CREATE CERTIFICATE [cert_ServerA]
AUTHORIZATION [MirroringUser_ServerA]
FROM FILE = 'C:\Temp\Mirroring\cert_ServerA.cer'
GO
- ☐ -- Grant Connection to the End-Point
GRANT CONNECT ON ENDPOINT::Endpoint_Mirroring TO [MirroringLogin_ServerA];
GO

2.4 On ServerB (connection from ServerC):

- ☐ -- Create Login for Witness (ServerC)
USE [master];
GO
CREATE LOGIN [MirroringLogin_ServerC] WITH PASSWORD = 'PWD1!';
GO
CREATE USER [MirroringUser_ServerC] FOR LOGIN [MirroringLogin_ServerC];
GO
- ☐ -- Create Certificate for Witness (ServerC)
CREATE CERTIFICATE [cert_ServerC]
AUTHORIZATION [MirroringUser_ServerC]
FROM FILE = 'C:\Temp\Mirroring\cert_ServerC.cer'
GO
- ☐ -- Grant Connection to the End-Point
GRANT CONNECT ON ENDPOINT::Endpoint_Mirroring TO [MirroringLogin_ServerC];
GO

2.5 On ServerC (connection from ServerA):

- ☐ -- Create Login for Mirror (ServerA)
USE [master];
GO
CREATE LOGIN [MirroringLogin_ServerA] WITH PASSWORD = 'PWD1!';
GO
CREATE USER [MirroringUser_ServerA] FOR LOGIN [MirroringLogin_ServerA];
GO
- ☐ -- Create Certificate for Mirror (ServerA)
CREATE CERTIFICATE [cert_ServerA]
AUTHORIZATION [MirroringUser_ServerA]
FROM FILE = 'C:\Temp\Mirroring\cert_ServerA.cer'
GO
- ☐ -- Grant Connection to the End-Point
GRANT CONNECT ON ENDPOINT::Endpoint_Mirroring TO [MirroringLogin_ServerA];
GO

2.6 On ServerC (connection from ServerB):

- ☐ -- Create Login for Mirror (ServerB)
USE [master];
GO
CREATE LOGIN [MirroringLogin_ServerB] WITH PASSWORD = 'PWD1!';
GO
CREATE USER [MirroringUser_ServerB] FOR LOGIN [MirroringLogin_ServerB];
GO
- ☐ -- Create Certificate for Mirror (ServerB)
CREATE CERTIFICATE [cert_ServerB]
AUTHORIZATION [MirroringUser_ServerB]
FROM FILE = 'C:\Temp\Mirroring\cert_ServerB.cer'
GO
- ☐ -- Grant Connection to the End-Point
GRANT CONNECT ON ENDPOINT::Endpoint_Mirroring TO [MirroringLogin_ServerB];
GO

3. Prepare Database

In this section you will restore the database in the mirror server. You must have the latest full backup file and log backup files of the database. The database will be restored in the mirror server with no recovery. The backup files must be accessible in the mirror server. This will prepare the restored database for the mirroring.

Reminder: disable any backup jobs before executing the manual backups.

Note: you must have the last log backup file in order to restore the database.

Note: if the database is set to Trustworthy, refer to the BOL for further information.

Note: if you are using Full-Text, refer to the BOL for further information.

You can use the following queries to have information on the backup file and check the current status of your progress:

- Query the backup file for information on the database and the database's files:

```
USE [master];
GO
RESTORE HEADERONLY FROM DISK = N'C:\Temp\Mirroring\myDatabase_FULL.bak';
GO
RESTORE FILELISTONLY FROM DISK = N'C:\Temp\Mirroring\myDatabase_FULL.bak';
GO
```

- Query for the database state (must be in Restoring status)

```
USE [master];
GO
SELECT [state_desc] FROM sys.databases WHERE [name] = 'myDatabase';
GO
```

3.1 On ServerA (connection from ServerC):

- ☐ -- Do a Full Backup
USE [master];
GO
BACKUP DATABASE [myDatabase] TO DISK = N'C:\Temp\Mirroring\myDatabase_FULL.bak' WITH INIT, STATS = 1;
GO
- ☐ -- Do a Log Backup
USE [master];
GO
BACKUP LOG [myDatabase] TO DISK = N'C:\Temp\Mirroring\myDatabase_LOG.trn' WITH STATS = 1;
GO

3.2 On ServerB (connection from ServerA):

- ☐ -- Restore Full Backup with NORECOVERY
USE [master];
GO
RESTORE DATABASE [myDatabase]
FROM DISK = N'C:\Temp\Mirroring\myDatabase_FULL.bak' WITH NORECOVERY
--, MOVE N'LogicalFileName' TO 'C:\DataFilesPath\PhysicalFileName' --<< Optionally
GO
- ☐ -- Restore Log Backup with NORECOVERY
USE [master];
GO
RESTORE LOG [myDatabase] FROM DISK = N'C:\Temp\Mirroring\myDatabase_LOG.trn' WITH NORECOVERY;
GO

4. Establish Database Mirroring

In this section you will create the mirroring.

4.1 On ServerB:

```
-- Set Principal in Mirror (local host: ServerB)
USE [master];
GO
ALTER DATABASE [myDatabase] SET PARTNER = 'TCP://ServerA:7024';
GO
```

4.2 On ServerA:

```
-- Set Mirror in Principal (local host: ServerA)
USE [master];
GO
ALTER DATABASE [myDatabase] SET PARTNER = 'TCP://ServerB:7024';
GO
ALTER DATABASE [myDatabase] SET WITNESS = 'TCP://ServerC:7024';
GO
```

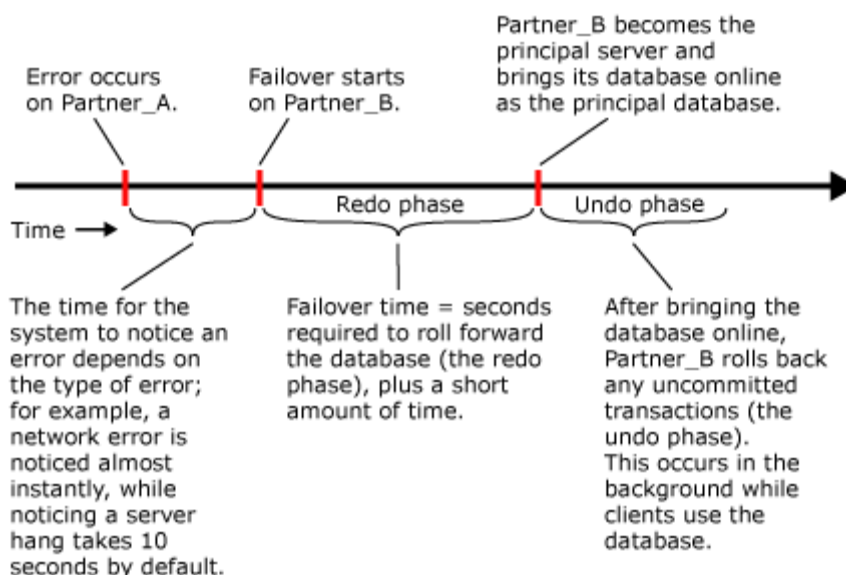
Post-Build

- ☐ Re-enable disabled backup jobs in the principal server.
- ☐ Check manually the failover forth and back, using the following script:

```
-- Manual Failover
USE [master];
GO
ALTER DATABASE [myDatabase] SET PARTNER FAILOVER;
GO
```

Failover

- Failover require at least 2 servers in the quorum, one of them is the mirror server.
- Automatic failover require a witness.
- During the failover, the database will not be accessible until fully recovered, as detailed in the image:



Monitoring

- ☐ Check the following performance counters: Transaction Delay, Redo Queue and Log Send Queue.
- ☐ Check the **expiration date** of the certificates.

- Database Mirroring Monitor is a monitoring tool accessible using the SQL Server Management Studio. This tool provides run-time information on the mirroring once you registered the mirrored database. You can find it under the mirrored database (right click on the database) *Tasks* → *Launch Database Mirroring Monitor*.
- Additional mirroring information, configuration and alerts can be accessible using the *sp_dbmmonitor...* stored procedures (dbm is used as prefix for database mirroring stored procedures).

Break Mirroring

- In some cases you would want to break the mirroring; such a case is when both the witness and principal servers not responding and quorum is lost, so the mirror database can replace the principal. In such cases you can break mirroring and set the database online.

Note: once mirroring is broken, it must be recreated again from the latest backup files (full + log).

```
-- Break Mirroring !!!
USE [master];
GO
ALTER DATABASE [myDatabase] SET PARTNER OFF;
GO
-- Get Mirror Database Online
USE [master];
GO
RESTORE DATABASE [myDatabase] WITH RECOVERY;
GO
```

Resources

[Troubleshoot Database Mirroring Configuration \(SQL Server\)](#)