**Disaster recovery: backing up the tail of the log**

<http://www.sqlskills.com/BLOGS/PAUL/post/Disaster-recovery-101-backing-up-the-tail-of-the-log.aspx>

One of the first things you should always check when a database has been damaged and you're going to perform a restore operation is whether you need to back up the tail of the log.

The tail of the log contains the transaction log that's been generated since the most recent log backup was taken. If you want to be able to recover right up to the point of the disaster, you need to be able to get those log records so they can be part of the restore sequence.

There are two cases to consider here 1) the server is still available 2) the server is not available. For case 1, you can just perform a regular tail-of-the-log backup. For case 2, you'll need to hack-attach the log into another server to be able to back it up.

**Case 1: tail-of-the-log backup when server is available**

If the database is damaged but the server is still available, taking a tail-of-the-log backup is simple. The only exception is when there has been a minimally-logged operation performed in the BULK\_LOGGED recovery model since the previous log backup - in that case a tail-of-the-log backup is not possible at all, and you'll have lost all transaction log generated since the last log backup. See [**A SQL Server DBA myth a day: (28/30) BULK\_LOGGED recovery model**](http://www.sqlskills.com/BLOGS/PAUL/post/A-SQL-Server-DBA-myth-a-day-(2830)-BULK_LOGGED-recovery-model.aspx) for more details.

When the data files are damaged and you want to take a log backup, you'll get an error if you try to back up the log normally.

As an example, I'll create a database and put some transactions in it:

CREATE DATABASE DBMaint2008;  
GO  
USE DBMaint2008;  
GO

CREATE TABLE TestTable (C1 INT IDENTITY, C2 CHAR (100));  
GO

-- Take a full backup  
BACKUP DATABASE DBMaint2008 TO DISK = 'D:\SQLskills\DemoBackups\DBMaint\_Full.bck' WITH INIT;  
GO

-- Insert some rows  
INSERT INTO TestTable VALUES ('Transaction 1');  
INSERT INTO TestTable VALUES ('Transaction 2');  
GO

-- Take a log backup  
BACKUP LOG DBMaint2008 TO DISK = 'D:\SQLskills\DemoBackups\DBMaint\_Log1.bck' WITH INIT;  
GO

-- Insert some more rows  
INSERT INTO TestTable VALUES ('Transaction 3');  
INSERT INTO TestTable VALUES ('Transaction 4');  
GO

Now disaster will strike - I'll do the following to simulate a disaster:

1. ALTER DATABASE DBMaint2008 SET OFFLINE
2. Delete the data file

If I try to then access the database, I'll get:

ALTER DATABASE DBMaint2008 SET ONLINE;  
GO

Msg 5120, Level 16, State 101, Line 1  
Unable to open the physical file "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\DBMaint2008.mdf". Operating system error 2: "2(The system cannot find the file specified.)".  
Msg 945, Level 14, State 2, Line 1  
Database 'DBMaint2008' cannot be opened due to inaccessible files or insufficient memory or disk space.  See the SQL Server errorlog for details.  
Msg 5069, Level 16, State 1, Line 1  
ALTER DATABASE statement failed.

So I'll try a normal log backup:

BACKUP LOG DBMaint2008 TO DISK = 'D:\SQLskills\DemoBackups\DBMaint\_Log\_Tail.bck' WITH INIT;  
GO

Msg 945, Level 14, State 2, Line 1  
Database 'DBMaint2008' cannot be opened due to inaccessible files or insufficient memory or disk space.  See the SQL Server errorlog for details.  
Msg 3013, Level 16, State 1, Line 1  
BACKUP LOG is terminating abnormally.

It doesn't work - as the data files aren't all accessible.

The trick is to use the NO\_TRUNCATE option, which allows the log backup even if the database files aren't there:

BACKUP LOG DBMaint2008 TO DISK = 'D:\SQLskills\DemoBackups\DBMaint\_Log\_Tail.bck' WITH INIT, NO\_TRUNCATE;  
GO

Processed 2 pages for database 'DBMaint2008', file 'DBMaint2008\_log' on file 1.  
BACKUP LOG successfully processed 2 pages in 0.010 seconds (0.972 MB/sec).

I can then use the tail-of-the-log backup as the final backup in the restore sequence, saving transactions 3 and 4 from above. Try it for yourself.

**Case 2: tail-of-the-log backup when server is no longer available**

This is the case where the server crashed and cannot be brought online. If you have access to all the data and log files for the database, you can attach it to another server and crash recovery will run automatically. If you only have access to the log file, you'll need to perform some extra steps to allow a tail-of-the-log backup to be performed - basically performing what I call a hack-attach.

Assuming I've run the script above to setup the database, this time I'll do the following to simulate a server-crash disaster:

1. ALTER DATABASE DBMaint2008 SET OFFLINE
2. Delete the data file
3. Copy the log file somewhere else
4. DROP DATABASE DBMaint2008

Now all I have is some backups and a log file. I'll need to attach the log file to SQL Server somehow so that I can perform the vital tail-of-the-log backup. The way to do it is:

1. Create a dummy database with the same name as the one that we're interested in (make sure you have instant file initialization enabled so the file creations don't take ages - see [**this blog post**](http://www.sqlskills.com/BLOGS/PAUL/post/A-SQL-Server-DBA-myth-a-day-(330)-instant-file-initialization-can-be-controlled-from-within-SQL-Server.aspx))
2. Set the database offline (or shutdown the server)
3. Delete all the files from the dummy database
4. Drop in the log file from our real database

Like so for steps 1 and 2:

CREATE DATABASE DBMaint2008;  
GO  
ALTER DATABASE DBMaint2008 SET OFFLINE;  
GO

Now I'll perform steps 3 and 4.

I need to attempt to bring the database online again:

ALTER DATABASE DBMaint2008 SET ONLINE;  
GO

Msg 5120, Level 16, State 101, Line 1  
Unable to open the physical file "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\DBMaint2008.mdf". Operating system error 2: "2(The system cannot find the file specified.)".  
Msg 945, Level 14, State 2, Line 1  
Database 'DBMaint2008' cannot be opened due to inaccessible files or insufficient memory or disk space.  See the SQL Server errorlog for details.  
Msg 5069, Level 16, State 1, Line 1  
ALTER DATABASE statement failed.

And then I can perform the tail-of-the-log backup as before, and use it to recover everything up to the point of the disaster.

**Note**: This procedure does not work if I try to perform the hack-attach to a more recent version of SQL Server. I tried hacking a 2005 log into a 2008 server and taking the tail-of-the-log backup - which worked fine, but the tail-of-the-log backup could not be used in conjunction with the first set of backups from the 2005 server. The reason for this is that the database version number in the tail-of-the-log backup is 655 (SQL Server 2008) and those for the 2005 backups are 611 (SQL Server 2005). The database doesn't get upgraded when restoring on the 2008 server until the end of the restore sequence - so as far as the 2008 server is concerned, my 2008 tail-of-the-log backup can't be used in the restore seqeunce of a still-2005-really database. Hope that makes sense!

**Summary**

Checking whether a tail-of-the-log backup is required is essential during a disaster recovery, and it's not hard to do. As with any disaster recovery procedures, make sure you've practiced doing it in advance!