**TechEd Demo: Creating, detaching, re-attaching, and fixing a suspect database**

This is one I've been trying to get to since I started blogging a couple of years ago: how to re-attach a detached suspect database. This is a pretty common scenario I see on the forums - a database goes suspect so the DBA tries to detach/attach, which fails. I wrote a demo for my corruption session at TechEd this year that shows how to create a suspect database with hex editor, then detaches it and shows how to re-attach and fix it. It's going to be a long blog post, but I've got everything wrapped up before we fly out to the UK tomorrow so I've got a bit of spare time.

**Creating a Suspect Database**

First off I'm going to create a simple database to use, called DemoSuspect with a table and some random data.

USE MASTER  
GO

CREATE DATABASE DemoSuspect  
GO

USE DemoSuspect;  
GO

CREATE TABLE Employees (FirstName VARCHAR (20), LastName VARCHAR (20), YearlyBonus INT);  
GO  
INSERT INTO Employees VALUES ('Paul', 'Randal', 10000);  
INSERT INTO Employees VALUES ('Kimberly', 'Tripp', 10000);  
GO

Now I'll perform an update in an explicit transaction and force it to be written out to disk with a CHECKPOINT. I've accidentally deleted Kimberly's bonus!

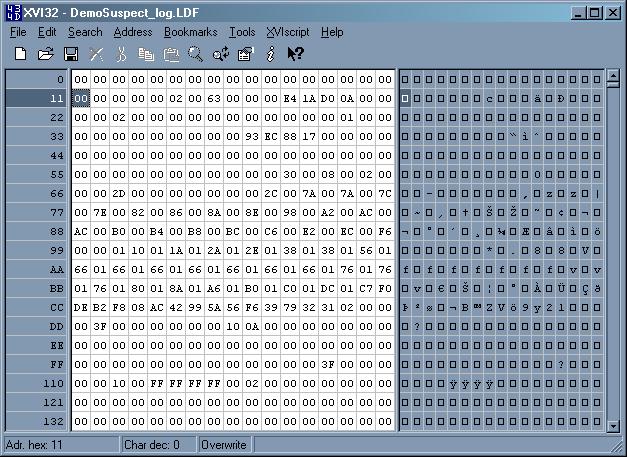
-- Simulate an in-flight transaction  
BEGIN TRAN;  
UPDATE Employees SET YearlyBonus = 0 WHERE LastName = 'Tripp';  
GO

CHECKPOINT;  
GO

Then in another window, I'll simulate a crash using:

SHUTDOWN WITH NOWAIT;  
GO

Now that SQL Server is shutdown, I'm going to simulate an I/O failure that corrupts the log file. I'm going to use a hex editor to do this - my editor of choice is the immensely popular and useful [**XVI32**](http://www.chmaas.handshake.de/delphi/freeware/xvi32/xvi32.htm), written by [**Christian Maas**](http://www.chmaas.handshake.de/delphi/freeware/freeware.htm). I opened the log file, filled the first section with zeroes, and then saved it again. See the screenshot below.



When I startup SQL Server again, it will try to run recovery on the DemoSuspect database and fail. This will put the database into theSUSPECT state.

So I restarted SQL Server, let's try getting in to the DemoSuspect database.

USE DemoSuspect;  
GO

Msg 945, Level 14, State 2, Line 1  
Database 'DemoSuspect' cannot be opened due to inaccessible files or insufficient memory or disk space. See the SQL Server errorlog for details.

Now let's check the database status:

SELECT DATABASEPROPERTYEX ('DemoSuspect', 'STATUS') AS 'Status';  
GO

Status  
--------  
SUSPECT

Now at this point, the correct procedure is to restore from backups. If there are no backups available, then the next best thing is to get the database into EMERGENCY mode and extract as much data as possible, or run emergency-mode repair. However, I'm going to try the detach/attach route instead.

**Detaching the Database**

I'll try detaching the database using sp\_detach\_db:

EXEC sp\_detach\_db 'DemoSuspect';  
GO

Msg 947, Level 16, State 1, Line 1  
Error while closing database 'DemoSuspect'. Check for previous additional errors and retry the operation.

Hmm - did it work or didn't it?

SELECT \* FROM sys.databases WHERE NAME = 'DemoSuspect';  
GO

And there are no results, so the detach must have succeeded.

**Re-attaching the Database**

Let's try the obvious sp\_attach\_db:

EXEC sp\_attach\_db @dbname = N'DemoSuspect',   
   @filename1 = N'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DemoSuspect.mdf',   
   @filename2 = N'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DemoSuspect\_LOG.ldf';  
GO

Msg 5172, Level 16, State 15, Line 1  
The header for file 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DemoSuspect\_LOG.ldf' is not a valid database file header. The PageAudit property is incorrect.

Hmm. How about using the ATTACH\_REBUILD\_LOG option on CREATE DATABASE? That should create a new log file for me:

CREATE DATABASE DemoSuspect ON  
   (NAME = DemoSuspect, FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DemoSuspect.mdf')  
FOR ATTACH\_REBUILD\_LOG;  
GO

Msg 5172, Level 16, State 15, Line 1  
The header for file 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DemoSuspect\_LOG.ldf' is not a valid database file header. The PageAudit property is incorrect.  
File activation failure. The physical file name "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DemoSuspect\_LOG.ldf" may be incorrect.  
The log cannot be rebuilt because the database was not cleanly shut down.  
Msg 1813, Level 16, State 2, Line 1  
Could not open new database 'DemoSuspect'. CREATE DATABASE is aborted.

Hmm. The database knows that there was an active transaction. Using the ATTACH\_REBUILD\_LOG command only works if the database was cleanly shut down and the log is missing. What about if I actually remove the log file and try that - maybe I could fool it? I took a copy of the data and log file, and then deleted the original log file (this is getting more and more dodgy...). Let's try again:

CREATE DATABASE DemoSuspect ON  
   (NAME = DemoSuspect, FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DemoSuspect.mdf')  
FOR ATTACH\_REBUILD\_LOG;  
GO

File activation failure. The physical file name "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DemoSuspect\_LOG.ldf" may be incorrect.  
The log cannot be rebuilt because the database was not cleanly shut down.  
Msg 1813, Level 16, State 2, Line 1  
Could not open new database 'DemoSuspect'. CREATE DATABASE is aborted.

Well, that was a long-shot but SQL Server's smarter.

Basically the problem is that the database wasn't cleanly shutdown, which means that recovery HAS to run and complete before the database can be attached again. Given that our log file is corrupt, that's impossible. So, **never detach a suspect database.**

The only way to get the database back into SQL Server is to use a hack. I'm going to create a new dummy database with the exact same file layout and as close as possible to the file sizes of the detached database. Then I'm going to shutdown SQL Server, swap in the corrupt database files, and re-start SQL Server. If all goes well, the corrupt, suspect database will be attached again.

The one major downside of this is that if the SQL Server instance doesn't have instant initialization enabled (see [**How to tell if you have instant initialization enabled?**](http://www.sqlskills.com/blogs/paul/2008/08/11/HowToTellIfYouHaveInstantInitializationEnabled.aspx)), then creating the dummy database could take a long time if the data files are very big. This means that your application is offline while the files are created and zero'd out.

So, I've already got a copy of the corrupt database files, so now I need to delete the data file too. Before doing this you want to make absolutely sure you've got multiple copies of the corrupt database files... just in case. After deleting the data file, I can create my dummy database.

CREATE DATABASE DemoSuspect  
GO

If you forget to delete the existing corrupt files first, you'll get the following error:

Msg 1802, Level 16, State 4, Line 1  
CREATE DATABASE failed. Some file names listed could not be created. Check related errors.  
Msg 5170, Level 16, State 1, Line 1  
Cannot create file 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\DemoSuspect.mdf' because it already exists. Change the file path or the file name, and retry the operation.

Ok - delete the data file and try again. Now I need to check the dummy database's files are there (at this point I'm totally paranoid), shutdown SQL Server, delete the dummy databases' files (this just seems wrong!), and swap in the corrupt files. I took another copy of the corrupt files before swapping them back in, just in case something goes wrong.

After restarting SQL Server, I can check the database state:

SELECT DATABASEPROPERTYEX ('DemoSuspect', 'STATUS') AS 'Status';  
GO

Status  
--------  
SUSPECT

Woo-hoo - I'm back to having a suspect database attached again - after having to shutdown the server and mess about deleting and copying files around. Not good. Now I can actually fix it.

**Fixing the Database**

If you don't have any backups, then the only way to get into the database is to use EMERGENCY mode. This lets you into the database but you need to be aware that recovery has not completed so the contents of the database are transactionally (and possibly structurally) inconsistent. I'm going to choose to repair the database using emergency-mode repair. See [**CHECKDB From Every Angle: EMERGENCY mode repair - the very, very last resort**](http://www.sqlskills.com/blogs/paul/2007/10/02/CHECKDBFromEveryAngleEMERGENCYModeRepairTheVeryVeryLastResort.aspx) for a detailed description of this tool.

ALTER DATABASE DemoSuspect SET EMERGENCY;  
GO  
ALTER DATABASE DemoSuspect SET SINGLE\_USER;  
GO  
DBCC CHECKDB (DemoSuspect, REPAIR\_ALLOW\_DATA\_LOSS) WITH NO\_INFOMSGS, ALL\_ERRORMSGS;  
GO

Msg 5172, Level 16, State 15, Line 1  
The header for file 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\DemoSuspect\_log.LDF' is not a valid database file header. The PageAudit property is incorrect.  
File activation failure. The physical file name "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\DemoSuspect\_log.LDF" may be incorrect.  
The log cannot be rebuilt because the database was not cleanly shut down.  
The Service Broker in database "DemoSuspect" will be disabled because the Service Broker GUID in the database (9E879BFC-B742-4A69-AB14-4D6BD6F99E02) does not match the one in sys.databases (B4568D23-7018-40CF-B189-9C29DE697C09).  
Warning: The log for database 'DemoSuspect' has been rebuilt. Transactional consistency has been lost. The RESTORE chain was broken, and the server no longer has context on the previous log files, so you will need to know what they were. You should run DBCC CHECKDB to validate physical consistency. The database has been put in dbo-only mode. When you are ready to make the database available for use, you will need to reset database options and delete any extra log files.

First off it tries to do the regular ATTACH\_REBUILD\_LOG. When that fails, DBCC CHECKDB takes over and forces the log to be rebuilt, after trying to force as much of the damaged log to be processed for recovery as it can. It then runs a full repair, in case there's anything corrupt in the database - in this case there isn't so there are no corruption messages in the output.

Notice the line about the Service Broker GUID being wrong. I had to use the hack method to get the database attached again, but when I created the dummy database, it created a Service Broker GUID for the DemoSuspect database in master.sys.databases. When I swapped in the corrupt database, it has a different GUID - so now I can't use Service Broker! And all because I detached the suspect database instead of repairing it.

So what's the state of the data?

USE DemoSuspect;  
GO

SELECT \* FROM Employees;  
GO

FirstName   LastName   YearlyBonus  
----------  ---------  ------------  
Paul        Randal     10000  
Kimberly    Tripp      0

Kimberly doesn't get a bonus this year - she won't be happy! This is contrived and flippant, of course, but it illustrates the point that after doing an emergency-mode repair, transactions that were active at the time the log was damaged will not get a chance to roll-back, most likely. In this case, I know what was going on when the crash occured, but what about on a busy OLTP system with hundreds or thousands of active transactions? What state will the data be in?

**Summary**

Yes, you can recover from a detached suspect database, but it's not pretty and you have to be very careful. The best course of action is always to have a comprehensive backup strategy that allows you to restore as quickly as possible. If you do have a suspect database and no backups, use EMERGENCY mode to access and/or repair the database. Hopefully this article will help people that find themselves in these situations.