# SQL Server: Advanced Corruption Recovery Techniques

Module 5: Advanced Restore Techniques

Paul S. Randal
http://www.SQLskills.com/blogs/paul/
Paul@SQLskills.com

pluralsight
hardcore developer training

## Introduction

- **Most people have executed a simple database restore but there are more advanced restore options that can be used**
- **Remember:**
  - Backups have to exist to be useful
  - Backups have to be valid to avoid data loss
- **In this module we'll cover:**
  - Tail-of-the-log backups when the original server is not available
  - Page and partial restores
  - Examining log backup contents
  - System databases

## What if SQL Server is Unavailable?

- **You may need a tail-of-the-log backup during disaster recovery**
- **If the SQL Server instance is unavailable, you'll have to perform a "hack-attach" of the log file to another instance to perform the backup**
  - Similar to hack-attaching a detached SUSPECT database
- **Steps to perform:**
  - Create dummy database with the same name
  - Set the dummy database offline
  - Delete all data and log files
  - Drop in the salvaged log file you want to back up
  - Try to bring the database online
  - Perform the tail-of-the-log backup
- **Note: this only works on the same version of SQL Server on which the original database existed**

## Page Restore

- **One or more pages can be restored**
  - Incredibly valuable if the database is not architected for partial database availability and you don’t want to restore the entire database
- **Backups must exist to allow the page to be restored to the same point in time as the PRIMARY filegroup**
- **The following page types cannot be page restored:**
  - Boot page
  - Fileheader pages
  - Allocation bitmaps (not including IAM pages)
  - Certain pages in critical system tables

## Piecemeal and Partial Restores

- **Uses partial database availability**
- **Example: you want to recover deleted data from a table but don’t want to restore the entire database**
  - Restore a subset of filegroups to create a new, smaller database
    - First restore must use the PARTIAL option and be the PRIMARY filegroup
    - Then roll forward to desired point in time
  - Manually extract deleted/damaged data from the restored database, then merge/insert it into the production database
    - Merge/insert is time-consuming and error-prone
- **Example: you want to restore a single corrupt filegroup**
  - Piecemeal restore into a database can be done online in Enterprise Edition
  - Setting the filegroup offline requires momentary exclusive database access

# Restoring Corrupt Backups

- **You may encounter a backup that is corrupt, but it's the only backup**
- **It can be restored using WITH CONTINUE_AFTER_ERROR**
- **Be careful about doing this with corrupt log backups**
  - They will create a corrupt database
- **If a backup in the middle of a restore sequence is corrupt, it may be better to end the restore sequence with the previous backup**
  - No corruption to deal with, but further back in time
- **Beware of tail-of-the-log backups that contain minimally-logged operations where the data file was not present**
  - Possible with SQL Server 2008 R2 onward
  - Guaranteed to corrupt the database during restore

## Looking Into Log Backups

- **Two undocumented table-valued functions exist to allow transaction log analysis**
- **fn_dblog for looking into the log**
  - select * from fn_dblog (startLSN, endLSN)
- **fn_dump_dblog for looking into log backups**
  - select * from fn_dump_dblog (startLSN, endLSN, 'DISK'|'TAPE', devicenum, backuppath, DEFAULT, DEFAULT,…)
- **When trying to find the exact time to restore to, fn_dump_dblog can vastly reduce the amount of time required**
  - E.g. finding the point at which a table was dropped
- **Can also be used for off-production analysis of activity**
- **Beware that each time fn_dump_dblog is used, it leaks a thread and a hidden thread scheduler**
  - Use on a test server, then use the Log Sequence Number (LSN) to do the restore in production

# Restoring to a Log Sequence Number

- **Once the desired LSN has been found, you usually want to restore everything up to, but not including, the transaction that begins there**
  - E.g. a transaction that does DROP TABLE or a large DELETE operation
- **Use RESTORE … WITH STOPBEFOREMARK='lsn:<lsnstring>'**
- **The hexadecimal LSN you get from the fn_dblog or fn_dump_dblog output needs to be translated into the form required by RESTORE**
  - I'll show you an example in the demo
  - The formula and code to do the translation are documented on my blog at http://bit.ly/10JT5b0 (those are zeros, not Os)

# Restoring System Databases (1)

- **You must use a backup from the same service pack level as the SQL Server instance**
  - E.g. an RTM backup of master will not restore on SP1
- **model**
  - Restore in same way as user databases
- **msdb**
  - Restore in same way as user databases
  - Stop the SQL Server Agent before restoring
  - If using log backups, make sure to restore them as well
- **mssqlsystemresource**
  - Not a real database and cannot be backed up or restored
  - Use file-system copy operations but be careful not to overwrite it with an older version
- **tempdb**
  - Cannot be backed up or restored

## Restoring System Databases (2)

- **Everyone should practice restoring master to see how easy it can be**
- **Restoring master**
  - Start SQL Server in single-user mode with the –m startup parameter
  - Restore the latest master backup, using WITH REPLACE
  - Server shuts down automatically, so remove –m and restart
  - For any changes after the master backup was made:
    - Reattach any new databases and recreate any new users/logins and server-level objects such as linked servers and server role memberships
- **Rebuilding master**
  - Necessary if no backup exists or the instance will not start
  - If you have to rebuild master to allow SQL Server to start, all data in the master database is lost
  - Restore master if possible using steps above or reattach all databases and recreate all users/logins and server-level objects
- **Books Online *Rebuild System Databases* at http://bit.ly/14nPPrM**

## Summary

- **Recovery time can be greatly reduced using:**
  - Page restore, as long as log backups exist to roll the page forward
  - fn_dump_dblog to find the point to which to restore the database
- **You should practice restoring the master database before you're forced to do it for real**

- **In the next module, we'll discuss:**
  - System table corruption
  - When not to run repair
  - Reconstructing deleted data
  - Manually editing data files