Back From the Dead: How to Restore a SQL Server in 60 Minutes or Less*

***Assuming You Plan for It**

Brad M. McGehee
SQL Server MVP
Director of DBA Education
Red Gate Software
www.bradmcgehee.com

So Many Ways Disasters Can Occur, and So Many Ways to Deal with Them

- There is no way I can cover every potential disaster scenario, unless I had several days to cover them all.
- Because of this, the focus of this session is very narrow, with an **emphasis on one of the most common and simple ways** to recover a server.
- I will be covering these assumptions as I proceed through the session.

My Assumptions About You

- You may be a part-time or full-time DBA.
- You may be a DBA Administrator or DBA Developer.
- You probably have less than 1-2 years experience working with SQL Server, but little experience restoring databases in the context of an emergency.
- You probably manage 10 or less SQL Server instances.
- Your databases are probably 100GB or smaller.
- You probably don’t have a 24/7 uptime requirement.

What are We Going to Learn Today

- Planning for the Worst
- Different Ways to Recover a SQL Server Instance
- How to Prepare for Database Recovery
- Introducing our Sample Disaster Scenario
- Demo a Safe and Easy Way to Recover a SQL Server Instance (Although it Requires Planning)

What We Won’t be Covering Today

- What we won’t be talking about:
  - Log Shipping
  - Database Mirroring
  - Failover Clustering or Virtual Server Failover
- All of these are good options, but more often than not in the real world, none of the above have been implemented for most SQL Server instances.
- The focus of this session is on **how do you recover your instance when the above options have not been implemented**, and all you have are backups.

Different Ways to Recover a SQL Server Instance

- Rebuild a new server from scratch.
- Rebuild a new server from an image.
- Cold backup SQL Server. (Not well prepared.)
- Warm backup SQL Server. (Well prepared & our assumption for our upcoming demo.)
- Use an existing SQL Server already in production.

- All of the above options are similar, but with minor variations that can affect how fast we can recover.

Planning for the Worst #1

- As DBAs, it is our job to protect the organization's data. More specifically, this means:
  - We realize that it is **not if, but when**, we will experience a problem that results in the loss of a database/server.
  - We know how to *properly* **backup** databases.
  - We know how to **restore** databases so that they can be back in production as quickly as possible, with the least amount of data loss possible.
  - But **the thing that keeps me up at night** is will I know how to deal with all the different combinations of problems that I could potentially face?

Planning for the Worst #2

- When databases go bad, there is often a cascading set of events that occur that often make the problem worse. **Never forget this**.
- As DBAs, we need to be prepared to deal with **as many situations** as possible.
- The only way to really prepare is to **produce a carefully thought-out DR plan** that covers as many contingencies as possible, and to practice them.
- A **major goal of this session** is to get you thinking about what should be added to your DR plan.

So How Do You Plan for the Worst?

- For the rest of this session, I am going to **demonstrate a safe and easy way to recover a user database/server** based on the scenario already stated.
- Keep in mind that this option assumes that you have done your homework and **you are prepared** to perform this task.
- **If you are prepared**, you will be up and running fast.
- **If you are not prepared**, you are setting yourself up for failure, a lot of stress, and unhappy users.

How to Prepare for Database Recovery #1

- Planning has gone into deciding **where databases should be moved** in case of a database failure.
  - Keep a spare server on hand (new hardware, cold, warm)
  - Use existing servers
- Planning has gone into determining the best way to **point your application** at a new SQL Server name.
- Full, differential (if used), and transaction log **backups are taken consistently, tested, stored away from the server immediately after being made, but are readily available** when needed.

How to Prepare for Database Recovery #2

- **Scripts have been prepared** to restore databases. Don’t depend using SSMS for restoring.
- All **database-related objects (the database ecosystem) have been scripted**, kept up-to-date, documented, and are available.
  - Security
  - Jobs
  - Linked Servers
  - SSIS Packages
- **All of the above has been documented** so that they can be followed in a step-by-step.

How to Prepare for Database Recovery #3

- **Instant file initialization has been turned on** for the servers to accept the restored databases.
  - http://www.bradmcgehee.com/2010/07/instant-file-initialization-speeds-sql-server/
- **Backup compression should** be used to make the backups & to speed the performance of restores.
- **Folder structures are standardized** for all SQL Servers so you don't have to worry about different file locations. If not, then ensure scripts can deal with this.
- **Test your plan** regularly to see if it works.

Assuming Planning Has Been Done Right

- We are now prepared for dealing with many (but of course not all) database recovery situations.
- So let’s take a look at our database recovery example.

Introducing our Sample Disaster Scenario #1

- A single SQL Server instance running on a server.
- Instance has system databases and only one user database. TDE is not being used.
- MDFs & LDFs reside on different arrays (local, DAS, SAN).
- SQL Server Agent jobs perform database backups and maintenance.
- Full backups are taken nightly. No differentials.
- Transaction log backups are taken every 15 minutes.
- Backups are moved immediately to other device.

Introducing our Sample Disaster Scenario #2

- The array that holds the MDF files for the system and user database begins to fail, and major corruption begins to occur on the **user database** due to hardware failure. Application can't access data.
- At this point, there are two major possibilities:
  - Database is not accessible at all, and a tail-log backup cannot be performed.
  - Database is partially accessible, and a tail-log backup can be performed to recover transactions that occurred since the last transaction log backup. This is our assumption.

Introducing our Sample Disaster Scenario #3

- We quickly realize that the existing hardware can’t be fixed immediately and the user database must be moved to a new server as soon as possible to minimize down time.
- The key now, is how fast can we get the database up and running on new hardware.
- How fast this is possible is dependent on how well prepared you are for dealing with such disaster scenarios.

Demo

- Explain primary server setup
- Explain secondary server setup
- Script users and jobs on primary
- Perform full, log, and tail-log backups on primary
- Restore AdventureWorks database on secondary
- Restore users and jobs on secondary
- Test secondary is working and application can talk to it.

Take Aways From This Session

- As the DBA, **it is your responsibility** to consider all the potential disasters you might encounter (**small to large**).
- You must **create a detailed and tested DR plan** to ensure that you can recover a database as quickly as possible and with the least amount of data loss as possible.
- **If you have not done so yet**, then make this your first priority when you get back to work.

E-Books, Websites, Slides & More

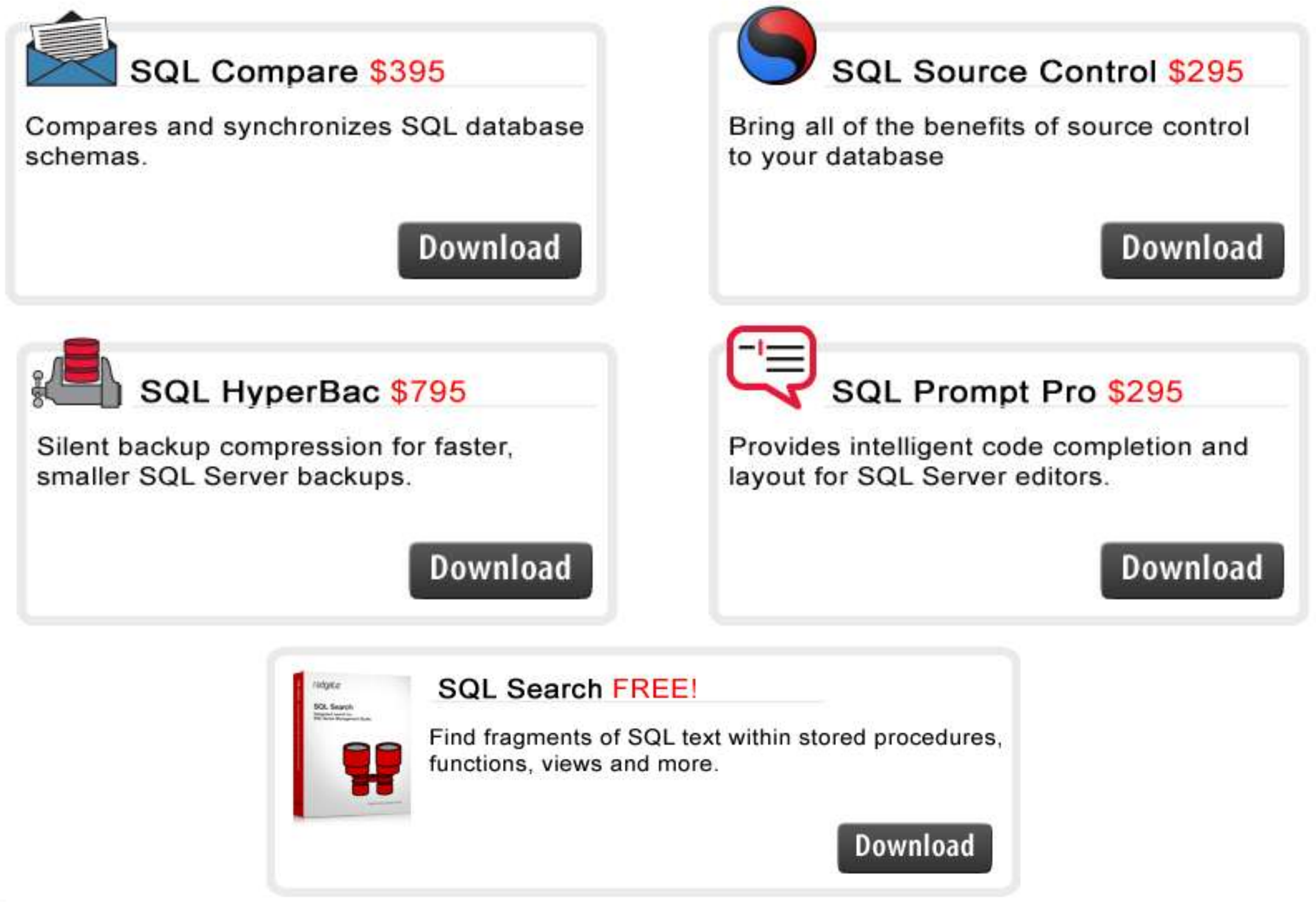
- Free E-books on SQL Server:
  - www.sqlservercentral.com/Books
  - http://www.sqlskills.com/blogs/paul/Commc
    nSQLServerMyths.pdf
- Check these websites out:
  - www.SQLServerCentral.com
  - www.Simple-Talk.com
- Blogs:
  - www.bradmcgehee.com
  - www.twitter.com/bradmcgehee
- Contact me at:
  - bradmcgehee@hotmail.com

How to Become an Exceptional DBA
Brad M McGehee
SQL Server Tacklebox
Rodney Landrum
Defensive Database Programming with SQL Server
Alex Kuznetsov
SQL Server Execution Plans
Grant Fritchey

redgate

"All the Red Gate products I used are a delight"

Doron Grinstein, Technical Director, Walt Disney



SQL Compare $395

Compares and synchronizes SQL database schemas.

SQL Source Control $295

Bring all of the benefits of source control to your database

SQL HyperBac $795

Silent backup compression for faster, smaller SQL Server backups.

SQL Prompt Pro $295

Provides intelligent code completion and layout for SQL Server editors.

SQL Search FREE!

Find fragments of SQL text within stored procedures, functions, views and more.

sponsored by

redgate

"I use SQL Compare every day. It's fast, helpful and reliable.
What more could I ask for?"
Matt DeWitt, Database Architect, 3M HIS

"Booyaa! Just finished adding all my databases to SVN using SQL Source Control. Amazing UI/UX experience."
Dave Dustin, SQL Server MVP

"Red Gate's tools are generally considered to be the gold standard....I wouldn't bother messing around with anything else."
Adam Machanic, SQL Server MVP