Background

Patient B's custom application had slowed down as their data grew. They'd tried several different relief efforts over time, but performance issues kept popping up – especially deadlocks. They wanted a permanent cure.

Let's see what we prescribed.

BRENT OZAR
UNLIMITED®

Patient B
SQL Critical Care®

AKA: "Why is this thing so slow?

What instance did we look at?

Instance: [REDACTED]

Applications involved: Primary SQL Server for all client applications

Total data size: 101GB

Memory Size: 64GB

Number of logical cores: 4

SQL Server version and edition: SQL Server 2012 SP2 Enterprise Edition (but licensed for Standard)

Virtualized? Yes (Hyper V)

Virtualization configuration

Lenovo ThinkServer RD630 –

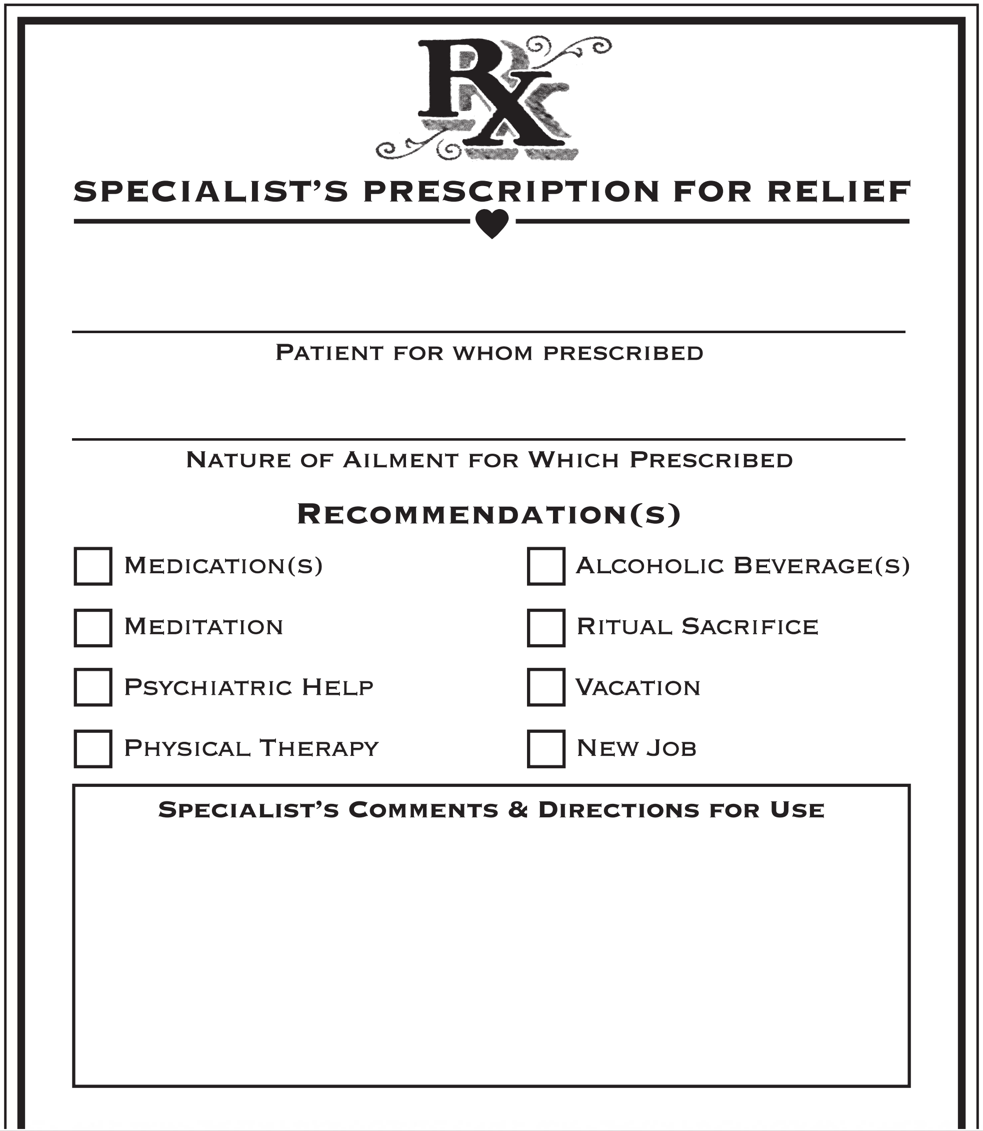
- 2 sockets, 8 cores each (+ HT)
- 256GB of RAM per host

4 hosts in cluster with identical RAM and CPU configuration

Hyper-V running on Windows Server 2012 R2

Dynamic load balancing, Dynamic memory enabled for this guest.

Executive Summary



Rx

SPECIALIST'S PRESCRIPTION FOR RELIEF

Patient for whom prescribed

Nature of Ailment for Which Prescribed

Recommendation(s)

□ Medication(s)

□ Meditation

□ Psychiatric Help

□ Physical Therapy

□ Alcoholic Beverage(s)

□ Ritual Sacrifice

□ Vacation

□ New Job

Specialist's Comments & Directions for Use

Your #1 pain point: Deadlocks and query performance

What your SQL Server told us…

In the last 40 days:

- Storage usually performs very well. You’re at or under Microsoft’s recommended latencies for reads and writes.
- The current amount of memory works well for your current query patterns (no need to go to Enterprise Edition on this account).
- 4 vCPUs is usually enough – but sometimes queries go parallel and use all 4 vCPUs
- Lock waits are moderate on average

Periodically you have a “poison” wait – codename RESOURCE_SEMAPHORE_QUERY_COMPILE

Solving deadlocks

We set up an Extended Events trace to capture deadlock graph and information

- Long term this is easier to use than ring buffer queries, plus it includes deadlock graphs

We saw two deadlocks the morning of 10/6– and both involved long multi-statement reports

- These reports need to have consistent data
- But because of some blocking pains in the past, NOLOCK joins are used against some tables

The reports are read only

- All inserts/updates are to temporary objects, not to shared tables

SNAPSHOT isolation to the rescue

When you allow SNAPSHOT isolation, you enable row versioning on the database

Then set long reports to use SNAPSHOT isolation

Two benefits:

- Deadlock Deterrent: The snapshot transaction will not block writers!
- You get consistent data from the beginning of a transaction to completion --- waaaay better than NOLOCK!

For your most deadlock prone databases, enable SNAPSHOT isolation, set up monitoring, and use it on reports

Can you handle the versioning workload?

Yes!

Your write levels and storage performance indicates you’re a very good fit

- tempdb write stalls at 2 ms (even for data files)
- < 1 GB total written to user database logs per day on average – and the highest two by far are [REDACTED] and [REDACTED] (which is a vendor database you can move elsewhere)

It’s also easy to enable snapshot and monitor performance first before you commit to using it

Very low writes

And you don’t have to create a version in tempdb for every single write

Example: for inserts there’s no previous version

A hidden transaction timestamp on the row itself means SQL Server doesn’t need the tempdb version

EVIDENCE
REDACTED

RESOURCE_SEMAPHORE_QUERY_COMPILE

There’s a backup on the compilation aisle

Compilation occurs when:

- A query can’t re-use a plan in cache
- A lot of data has changed, and SQL decides it’s best to get a new plan
- Or it has a hint forces it

But SQL Server can only compile a certain amount of queries at once

On your instance 16 “small” queries, 4 “medium” queries, or 1 “large” query

http://blogs.msdn.com/b/support_sql_france/archive/2012/02/07/sql-server-compilation-gateways-and-resource-semaphore-query-compile.aspx

How much does this slow you down?

There were 2.4 hours of this total

We saw 12 minutes of this wait between Monday and Tuesday in a 24 hour period

It doesn’t happen constantly, and the average wait time is low

But when this type of wait gets bad, it can make users feel like the instance isn’t working

That’s because their query can’t even get started

The plot thickens: the long compiler

You log the execution time of your reports

- We found a case where your logged time was 7 seconds longer than the query execution time

The difference: COMPILATION TIME

- This query has 27+ joins/subqueries
- SQL Server is giving it full compilation and taking up to 10 seconds to do so

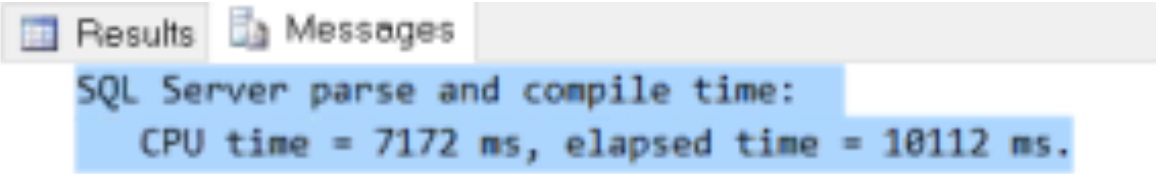
This query is definitely a “big” compile gate query (all those joins!)

So if more than one of these needs to compile at once.... RESOURCE_SEMAPHORE_QUERY_COMPILE

How we measured compile time

The simple method when you’re testing a query:

SET STATISTICS TIME ON;



```
SQL Server parse and compile time:
   CPU time = 7172 ms, elapsed time = 10112 ms.
```

For other queries against production, you need to shred execution plans in cache to see the compile time.

REDACTED
(3 slides of specific query recommendations)

Recap: how to solve your performance pain

Solve deadlocks

- Short term: indexing
- Long term: SNAPSHOT isolation for reports

Speed up slow reports

- Short term: indexing (BOGO deal here)
- Long term: Monitor compilation times, break “slow compiler” queries into smaller pieces

Long term scalability

- During busy periods, multiple users running reports will be tricky with just 4 vCPUs
- Consider SQL Server 2014 Standard Edition with 6 vCPUs long term (not required with current load, but this is not the busy season)
- Remove Dynamic Memory from the VM configuration

One note on HA/DR

A gap in your availability plan

Virtualization protects against hardware failures

SQL Server log backups help you recover data if there’s corruption or an “oops” delete

But what if …

- The VM’s OS drive fills up
- Windows in the VM starts bluescreening repeatedly
- The SQL Server instance starts stack dumping constantly and can’t be used

Virtualization doesn’t help with this

What’s the Plan B for this scenario?

What would you use to restore?

How much data loss would be involved?

How long would you be down?

Raise this up to management and talk about options and their licensing costs / performance overhead

If you need to close the gap...

Mirroring and AGs aren’t a good fit for your number of databases, your team (no fulltime DBA), your number of cores, or your licensing budget

Transaction Log Shipping could help

- Low performance impact – uses log backups you need to do anyway
- Works with Standard Edition
- Vendor tools like Dell LiteSpeed make setup and management simpler with your number of databases

No automatic failover, but helps narrow the gap for downtime and data loss

SQL Server Version and Dynamic Memory

Don’t use Dynamic Memory in Hyper-V

You want this in case of an emergency, but it makes your life more complicated

Microsoft recommends that you use a “Locked Pages” setting in SQL Server if you use this (otherwise the SQL Server can be paged out under memory pressure)

But when SQL Server comes under memory pressure, the chances of an OS crash go up when Locked Pages in enabled

Don’t enable Dynamic Memory - plan capacity so you never have to use it instead

Why SQL Server 2014?

This isn't absolutely necessary, but it has some perks

Enhanced tempdb IO performance – data is less "eager" to be flushed to disk in tempdb data files

- Since I'm recommending snapshot isolation/row versioning, reducing other IO in tempdb is attractive (but your storage does look up to the task)

Option for 128GB of memory if you want it in Standard Edition

- SQL Server isn't crying out for this right now, and indexes we're adding will reduce IO as well

Summary: this is nice to have if it doesn't cost you much different in licensing

Index Tuning

You have a good basis

Your strategy is to keep indexes consistent across all client databases (and you've got hundreds)

You have good clustered indexes

You have a lot of really narrow nonclustered indexes

You need to add "covering" indexes where it really matters

- I've got a set of these in your index prescription file

REDACTED
(4 slides of specific index recommendations)

Code for specific index changes are in your Index Prescription.sql file

Get around Deadlocks with Snapshot Isolation

REDACTED
(53 slides of training on
the right isolation levels to
fix their deadlock issues)

Other topics we discussed

“Slow in the Application, Fast in SSMS?”
http://www.sommarskog.se/query-plan-mysteries.html

Differences in temp tables and table variables (statistics, row estimates, and more)
http://www.brentozar.com/archive/2014/06/temp-tables-table-variables-memory-optimized-table-variables/

Your Action Plan

Your Action Plan is in a separate FILE

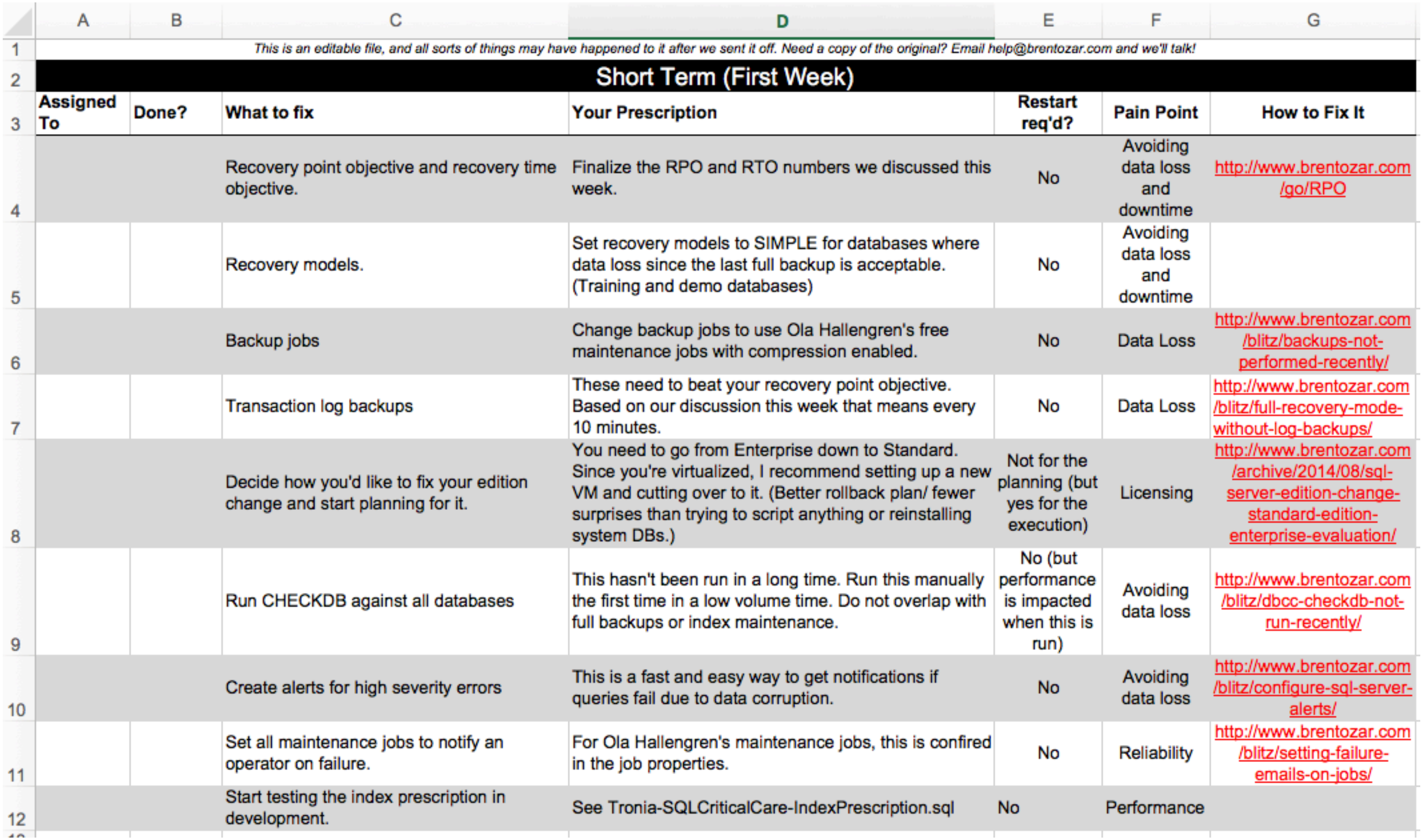
It's in Excel so you can assign folks and edit it

We're going to show screenshots of it here

If you're reading this later and don't have a copy, just email us at help@BrentOzar.com

(Don't worry, we'll make sure it's really OK to send it to anyone who asks for it.)

The action plan doesn't include AG architecture specifics, since you have to make the 1/2/3 choice first in order to build a project plan for that move.



This is an editable file, and all sorts of things may have happened to it after we sent it off. Need a copy of the original? Email help@brentozar.com and we'll talk!

Short Term (First Week)

Assigned To	Done?	What to fix	Your Prescription	Restart req'd?	Pain Point	How to Fix It
		Recovery point objective and recovery time objective.	Finalize the RPO and RTO numbers we discussed this week.	No	Avoiding data loss and downtime	http://www.brentozar.com/go/RPO
		Recovery models.	Set recovery models to SIMPLE for databases where data loss since the last full backup is acceptable. (Training and demo databases)	No	Avoiding data loss and downtime	
		Backup jobs	Change backup jobs to use Ola Hallengren's free maintenance jobs with compression enabled.	No	Data Loss	http://www.brentozar.com/blitz/backups-not-performed-recently/
		Transaction log backups	These need to beat your recovery point objective. Based on our discussion this week that means every 10 minutes.	No	Data Loss	http://www.brentozar.com/blitz/full-recovery-mode-without-log-backups/
		Decide how you'd like to fix your edition change and start planning for it.	You need to go from Enterprise down to Standard. Since you're virtualized, I recommend setting up a new VM and cutting over to it. (Better rollback plan/ fewer surprises than trying to script anything or reinstalling system DBs.)	Not for the planning (but yes for the execution)	Licensing	http://www.brentozar.com/archive/2014/08/sql-server-edition-change-standard-edition-enterprise-evaluation/
		Run CHECKDB against all databases	This hasn't been run in a long time. Run this manually the first time in a low volume time. Do not overlap with full backups or index maintenance.	No (but performance is impacted when this is run)	Avoiding data loss	http://www.brentozar.com/blitz/dbcc-checkdb-not-run-recently/
		Create alerts for high severity errors	This is a fast and easy way to get notifications if queries fail due to data corruption.	No	Avoiding data loss	http://www.brentozar.com/blitz/configure-sql-server-alerts/
		Set all maintenance jobs to notify an operator on failure.	For Ola Hallengren's maintenance jobs, this is confired in the job properties.	No	Reliability	http://www.brentozar.com/blitz/setting-failure-emails-on-jobs/
		Start testing the index prescription in development.	See Tronia-SQLCriticalCare-IndexPrescription.sql	No	Performance	

This is an editable file, and all sorts of things may have happened to it after we sent it off. Need a copy of the original? Email help@brentozar.com and we'll talk!

Medium Term (This Month)

Assigned To	Done?	What to fix	Your Prescription	Restart req'd?	Pain Point	How to Fix It
		Deadlocks	Begin testing SNAPSHOT isolation	No	Deadlocks	See your findings PDF for information and how to monitor it
		Database page verification setting	Change the page_verification setting in each database to CHECKSUM (helpful script in the URL to the right)	No	Avoiding data loss	http://www.brentozar.com/blitz/page-verification/
		Remove old CDW (copy database wizard) jobs	Have not run since 2014, no loner needed.	No	Simplifying management	
		Set all agent jobs to be owned by SA	When you create a job, the owner defaults to who runs the script-- and if it can't validate the owner in AD, the job can fail.	No	Reliability	http://BrentOzar.com/go/owners
		Use Ola Hallengren's backup history cleanup job.	When backup history never gets cleaned up, MSDB can grow to multiple GB and waste memory and resources.	No	Performance	http://BrentOzar.com/go/history
		Gradually introduce index maintenance	Use Ola Hallengren's scripts with the thresholds from the findings PDF.	No	Performance	See your findings PDF for specific commands
		Remove Dynamic Memory from your guest	Assign dedicated memory so you don't have to worry about SQL Server paging out under pressure	Yes	Performance	

SPECIALIST'S PRESCRIPTION FOR RELIEF

During our SQL Critical Care® together this week, we came up with a set of recommended changes to your server.

We've compiled them into a handy prescription we're handing over in editable files that are friendly for markup and copy-paste.

SCRIPTS AND COMMANDS MAY NEED CUSTOMIZATION.
THEY WILL NEED TESTING.
THEY MAY CAUSE ITCHING AND DISCOMFORT.
EDITABLE FILES MAY BE EDITED AFTER WE SEND THEM TO YOU.

Read through all recommended steps carefully, understand what we're suggesting, and test them.

Ideally, measure your server to know that these changes helped, and consider backing them out if they don't help.

If time has passed and you want a fresh copy of those editable files, email help@brentozar.com and we'll ask you for a secret password.

We believe this is how to solve your pain point.

Which task will be the most difficult for you?