[Template] Targeted Database Migration Checklist: <client name>

1. Initial Audit

Click here to expand...

A. Source Server - <hostname><IP>

I. SQL Server Level Settings

	Task	Pending	Notes
1	If non-MI DBs exist, run by customer		
2	If disabled jobs exist, run by customer		`SELECT @@servername as ServerName, name as JobName, date_modified as LastModifiedDate FROM msdb.dbo.sysjobs WITH (NOLOCK) WHERE enabled = 0 ORDER BY name`
3	If processor affinity settings are vanilla, check-off 1B-II-2		
4	If Dashboard objects do not exist, check-off section 2b		
5	If non-MI SSIS & SSAS objects exist, run by customer		
6	If IP and hostname will be changed post the migration, uncheck 2A-I-12 and 3-29		

II. DB Level Settings

	Task	Pending	Notes
1	If [medical] does not “Auto Shrink”, check-off “Day Of” Task #14		
2	Ensure [medical_test] uses simple recovery with TLs shrunk		
3	Ensure `PAGE_VERIFY CHECKSUM` is enabled <https://docs.microsoft.com/en-us/troubleshoot/sql/admin/troubleshoot-dbcc-errors#investigate-root-cause-for-database-consistency-errors >		

B. Target Server - <hostname><IP>

I. Windows Server Level Settings

	Task	Pending	Notes
1	☐ Ensure that the same services/programs exist (ex. SSIS, SSAS, SSDT for Visual Studio)	☐	Ensure that SQL Server Agent and SQL Server Browser has “automatic” startup type Here is a Microsoft KB for installing SSDT for Visual Studio: https://docs.microsoft.com/en-us/sql/ssdt/download-sql-server-data-tools-ssdt?view=sql-server-ver15 Here is the download page for Visual Studio’s “SSIS Projects” extension, which is also needed to configure the dtsx packages: https://marketplace.visualstudio.com/items?itemName=SSIS.SqlServerIntegrationServicesProjects Here is a Microsoft KB for installing SSIS: https://docs.microsoft.com/en-us/sql/integration-services/install-windows/install-integration-services?view=sql-server-ver15
2	☐ Ensure that Windows has been activated	☐	
3	☐ Run services in target with service acct as in source	☐	As reference, this article lists the permissions recommended by Microsoft: https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-windows-service-accounts-and-permissions?view=sql-server-ver15#Serv_Perm
4	☐ Ensure that network protocols are same as source	☐	If changes are made, restart the SQL Server service
5	☐ Confirm if server will be used for SQL Server only, or for the main app as well	☐	
6	☐ Ensure enough memory, CPU, storage, & drives were provisioned. See if dedicated drives (not C:\) can be provided for both the data and backups.	☐	If changes are made, reconfigure SQL Server accordingly (ex. set default backup directory, or set max server memory) **Note**: • If the server will be used for the main app as well, make sure it has enough space to house the MEDINFO folder as well. • If the server will be used for the main app, recommend that the drive names are the same
7	☐ Ensure firewall settings are same as source. If it is off in source but on in target, add inbound & outbound rules to open SQL Server ports (TCP 1433 & 1434, UDP 1434)	☐	https://msdn.microsoft.com/en-us/library/cc646023.aspx

II. SQL Server Level Settings

	Task	Pending	Notes
1	☐ Verify MSSQL 2012+ with latest CU is installed	☐	https://docs.microsoft.com/en-us/sql/database-engine/install-windows/latest-updates-for-microsoft-sql-server?view=sql-server-ver15 https://sqlserverbuilds.blogspot.com/
2	☐ Ensure that processor affinity settings are the same as source	☐	
3	☐ Ensure that mixed authentication is enabled	☐	

2A. Prepwork

Click here to expand...

I. SQL Server Level Settings

	Task	Pending	Notes
1	☐ Copy latest full backup and create dummy DB(s). If non C:\ drives were provisioned in step 1B-I-6, configure instance DB directories accordingly first.	☐	
2	☐ Recreate Logins and configure as in source	☐	Disable [sa] if [sa] is disabled in source `grant select, insert, update, delete to mwuser` `grant exec to mwuser` If creating Logins from scratch, create [usr] Login and tie with [mwuser] User
3	☐ Recreate jobs (excluding Maintenance Plans), then disable them	☐	`USE MSDB;` `GO` `UPDATE MSDB.dbo.sysjobs` `SET Enabled = 0` `WHERE Enabled = 1;` `GO` If creating jobs from scratch, here is the list of standard jobs: 1. "job_cldictionary_archive" 2. "Cleanup CdsRsnRequests [medical]" (omit for v75) 3. "Appt Search Scrub Data/ Build Cache" (omit for v75) 4. "Dimrun Cutoff (month to date) [medical]" 5. "Dimrun Collections [Medical]" 6. "Import NC Files" (contingent on eRX module evaluation) 7. Patient Portal Job(s) (contingent on Patient Portal module evaluation)
4	☐ Recreate Maintenance Plans	☐	If creating Maintenance Plans from scratch, can use Ola Hallengren maintenance solutions. For reference, here is a sample schedule: 1. Integrity check for user DBs every Sat at 2am 2. Integrity check for system DBs every Sat at 3am 3. Optimize index for [medical] every Sun at 2am 4. Full backup for [medical] every day at 10pm 5. Log backup for [medical] every hour of every day from 6am to 9pm 6. Cleanup command log every Sat at 12am 7. Cleanup output logs every Sun at 12am 8. Cleanup backup history every Sat at 1am 9. Cleanup job history every Sun at 1am
5	☐ Ensure server-level objects are same as source. If no proxy, check-off 2B-1	☐	

6	☐ Enable “Optimize for Ad Hoc”, “Compress Backup”, “xp_cmdshell” settings. ☐ If “Cost Threshold for Paralellism” is 5, set to 50. ☐ Set MAXDOP to 8 or lower <https://littlekendra.com/2016/07/14/max-degree-of-parallelism-cost-threshold-for-parallelism/> ☐ Ensure instant file initialization (IFI) and Lock Pages in Memory (LPIM) are enabled	☐	`sp_configure 'show advanced options', 1` `go` `reconfigure` `go` `sp_configure 'xp_cmdshell', 1` `go` `reconfigure` `go` <https://www.brentozar.com/blitz/instant-file-initialization/ > <https://blog.sqlauthority.com/2019/09/16/sql-server-enable-lock-pages-in-memory-lpim/ >
7	☐ If Patient Portal exists, enable “Database Mail XPs”	☐	
8	☐ Set Min Server Memory to 0 and Max Server Memory to what’s recommended by this script: https://github.com/bornsql/scripts/blob/main/max_server_memory.sql	☐	Note: If server is used to house both the app and database, lower the max memory by another 2GB for MedInformatix
9	☐ Check for "orphaned" users. If none, check-off 3-12.	☐	`EXEC sp_change_users_login 'REPORT'` `Use [$databaseName];` `GO` `ALTER USER OrphanUser WITH LOGIN = correctedLoginName` `SELECT *` `FROM sys.objects` `WHERE schema_id = SCHEMA_ID('dbo')`

10	☐ Setup and test “Import NC Files” job	☐	`begin tran` `select * from clparms as [a]` `where 1=1 and a.code='erx' and a.` `skey='up'` `update clparms set ALPHA1='D:` `\MEDINFO\ERX' where 1=1 and` `code='erx' and skey='up'` `update clparms set ALPHA3='D:` `\MEDINFO\' where 1=1 and` `code='erx' and skey='up'` `select * from clparms as [a]` `where 1=1 and a.code='erx' and a.` `skey='up'` `rollback tran` `--commit tran` `exec util_newcrop_import` `@filename='NCTSV-201907.EXE',` `@checkSyslog = 0, @debug = 1  --` `using last month's file`
11	☐ (Time permitting) Run DMA (Data Migration Assistant)	☐	
12	☑ Note down which objects are configured using IP or hostname	☐	
13	☐ Ensure there are 8 equally sized data files for tempdb. If there are less than 8 cores, create the same number for files as cores. If adding files, restart SQL Server. ☐ If on 2014 and below, enable TF 1117 and 1118	☐	https://wiki.galenhealthcare.com/index.php/Create_Multiple_TempDB_files_for_best_performance `DBCC TRACEON(1117, -1);` `DBCC TRACEON(1118, -1);`

II. 3rd Party Settings

	Task	Pending	Notes
1	☐ If Patient Portal exists, try configuring DB mail or reach out to Support	☐	Confirm whether a maintenance splashscreen should be implemented
2	☐ Check if customer uses reporting platform requiring ODBC connection	☐	
3	☐ If Interfaces exist, touch bases with Interface	☐	
4	☐ If HDL exists, touch bases with HDL engineer	☐	

5	☐ If Phone Tree exists, touch bases with Phone Tree engineer	☐	
6	☐ If Provider Portal exists, ensure that connection to the new DB server is tested	☐	Confirm whether a maintenance splashscreen should be implemented
7	☐ If Hef and/or RWT Exports exist, ensure Proxy, Credential, and export directory are copied over. Also, ensure that an engineer is assigned to re-install and test Box Sync post the migration.	☐	

2B. Dashboard Tasks

☐ Premium Dashboard not used. Disregard section

Click here to expand...

	Task	Pending	Notes
1	☐ Create SSIS proxy (if used)	☐	
2	☐ Migrate SSAS DB(s)	☐	
3	☐ Reconfigure, reimport, and test SSIS package(s)	☐	Custom packages may require that certain directories be moved over

3. The Day Of

Click here to expand...

	Task	Notes
1	☐ If Portal(s) exist, coordinate a time to implement splashscreen	
2	☐ Contact client 30 mins prior	
3	☐ At specified time, make sure Medinfo users are off	
4	☐ Disable jobs in source server and disable SQL Server Agent	Note which jobs are disabled `USE MSDB;` `GO` `UPDATE MSDB.dbo.sysjobs` `SET Enabled = 0` `WHERE Enabled = 1;` `GO`

5	☐ Place DB(s) in “read-only” mode	Use GUI if other users are connected `USE [master]` `GO` `ALTER DATABASE [dashboardDB] SET  READ_ONLY` `WITH NO_WAIT` `GO`
6	☐ Take “copy only” backup with “verify backup integrity”	
7	☐ Transfer backup(s) to new server	
8	☐ Drop “bogus” DB(s) in new server	
9	☐ Restore backup(s) to new instance	
10	☐ Set DB(s) from “read-only” to “read-write”	
11	☐ Check Login Permissions	`grant select, insert, update, delete to mwuser` `grant exec to mwuser`
12	☐ Check for "orphaned" users	`EXEC sp_change_users_login 'REPORT'` `Use [$databaseName];` `GO` `ALTER USER OrphanUser WITH LOGIN =` `correctedLoginName` `SELECT *` `FROM sys.objects` `WHERE schema_id = SCHEMA_ID('dbo')`
13	☐ Shrink logs (if autogrowth has kicked in)	

14	☐ Turn off auto shrink (if used)	
15	☐ Ensure that TL autogrowth settings are optimal	
16	☐ Setup and test “Import NC Files” job	 ``` begin tran select * from clparms as [a] where 1=1 and a. code='erx' and a.skey='up' update clparms set ALPHA1='D:\MEDINFO\ERX' where 1=1 and code='erx' and skey='up' update clparms set ALPHA3='D:\MEDINFO\' where 1=1 and code='erx' and skey='up' select * from clparms as [a] where 1=1 and a. code='erx' and a.skey='up' rollback tran --commit tran exec util_newcrop_import @filename='NCTSV- 201907.EXE', @checkSyslog = 0, @debug = 1 -- using last month's file ```
17	☐ Ensure server computer name has been renamed	https://msdn.microsoft.com/en-us/library/ms143799.aspx Default Instance: ``` sp_dropserver <old_name>; GO sp_addserver <new_name>, local; GO ``` Named Instance: ``` sp_dropserver <old_name\instancename>; GO sp_addserver <new_name\instancename>, local; GO ``` Verify: ``` SELECT @@SERVERNAME AS 'Server Name'; ``` If changed, SQL Server will have to be restarted.

18	☐ Point MI to new SQL Server, being sure to restart the Net Services and Redirector Service	If MSETUP is using “sa”, replace with “usr” If IP is being changed, make sure to use hostname instead
19	☐ Conduct unit testing	`select * from clmaster where plname='test' and account=`
20	☐ Re-enable jobs and restart SQL Server Agent	`USE MSDB;` `GO` `UPDATE MSDB.dbo.sysjobs` `SET Enabled = 1` `WHERE Enabled = 0;` `GO`
21	☐ Take full backup of [medical]	
22	☐ Set old DB(s) to offline (be sure that current Login’s default DB is not [medical])	`ALTER DATABASE <dbname> SET OFFLINE WITH ROLLBACK IMMEDIATE`
23	☐ Run dbcc checkdb	`dbcc checkdb('medical') WITH NO_INFOMSGS, ALL_ERRORMSGS`
24	☐ Rebuild indexes	
25	☐ Review Error Logs and SQL Server Logs for any errors	
26	☐ Remove tmp backups in target and source	
27	☐ If Portal exist(s), have Portal engineer remove splashscreen(s) and possibly reconfigure the DB connection	

28	☐ Confirm with customer that migration has been completed	
29	☑ After IP and/or hostname is changed, reconfigure affected objects	
30	☐ Update SF connect info, SQL Server Version, & SQL DB Server Version	

4. Baselining and Performance Tuning

Click here to expand...

Traditionally, this goes as follows:

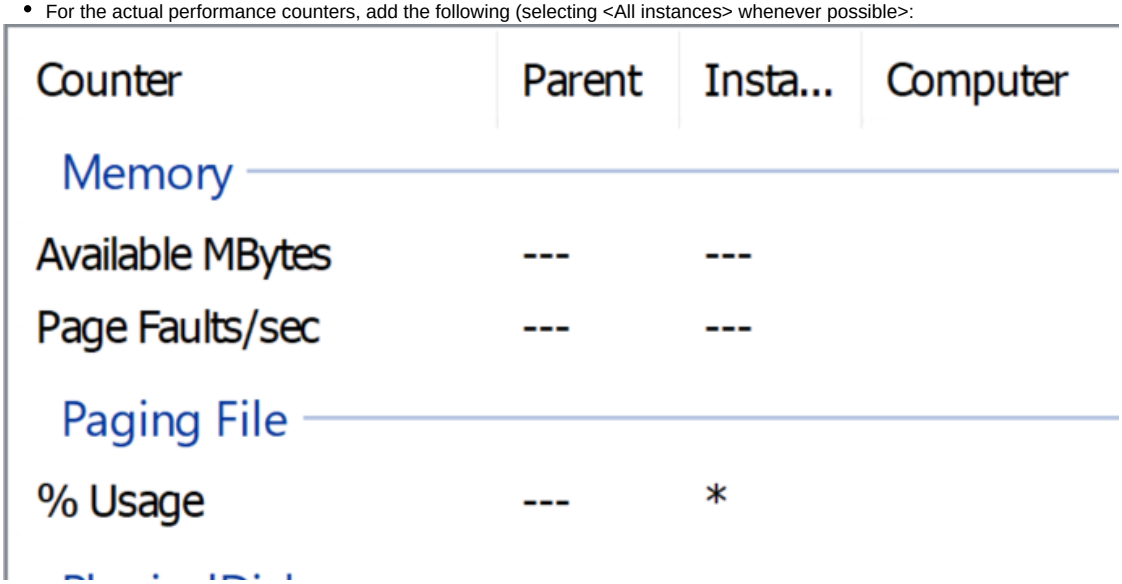
- Change the Compatibility Level to the latest version
- Collect perfmon counters
- Generate a PAL report and analyze its results
- If any alarming patterns are found, a trace is run during those times to see if they are caused by slow performing queries.

When crossing the SQL Server 2014 threshold, however, Microsoft recommends the following:

- Keep the source Compatibility Level
- Enable Query Store to collect baseline data
- If using SSMS v18+, enable Query Tuning Assistant
- Change Compatibility Level to latest version
- Fix performance regressions with Automatic Plan Correction (SQL 2017+)

A. Collect PerfMon Counters

I. Setup

	Task	Pending	Notes
1		☐	Recommended Settings: • For name, use “SQL Server Collector” and select "Create manually (Advanced)" • Select "Create data logs" and only check "Performance counter” • For interval, use 30 seconds at most. Per Scott Whigham, 30 is generally good enough but Brent Ozar recommends going less possible • For the actual performance counters, add the following (selecting <All instances> whenever possible>: 

In “PerformanceMonitor”, start a User Defined “Data Collector Set”

PhysicalDisk

% Disk Time	---	*
Avg. Disk Queue Length	---	*
Avg. Disk sec/Read	---	*
Avg. Disk sec/Write	---	*
Current Disk Queue Le...	---	*
Disk Reads/sec	---	*
Disk Writes/sec	---	*

Processor

% Processor Time	---	*

SQLServer:Buffer Manager

Page life expectancy	---	---

SQLServer:General Statistics

User Connections	---	---

SQLServer:Memory Manager

Memory Grants Pending	---	---

SQLServer:SQL Statistics

Batch Requests/sec	---	---
SQL Compilations/sec	---	---
SQL Re-Compilations/...	---	---

System

Processor Queue Length	---	---

			• Use the default directory for saving the data (default is %systemdrive%\PerfLogs\Admin\SQL Server Collector) • Select “Start this data collector set now” and click “Finish”

II. Analysis

	Task	Pending	Notes
1	☐ Stop the Data Collector Set from part I and move the file to a computer with PAL installed <https://github.com/clinthuffman/PAL >	☐	
2	☐ Generate a PAL report with the data collected	☐	Recommended Settings: • Do not restrict to a time range • For the Threshold File, select the Title with the latest SQL Server version available • Under questions: • Note down the server’s PhysicalMemory and OS • If server OS is newer than what' available, select “Windows Server” • For OLTPvsOLAP, this is usually “True” as [medical] is mostly written into and not used for data warehousing • For UsingInMem, this is usually “False” as we haven’t used In-Memory tables for [medical] • If unsure, leave as default • For analysis interval, leave as AUTO • In the Execute tab, select “Execute as a low priority process”
3	☐ Analyze PAL report and document analysis results	☐	Sample Analysis: `# Memory - Not ideal that paging file has nontrivial usage, but still OK` `- Available MBytes` `  - On average, there is at least 3 GB of memory available, so we should be fine (B.Ozar recommends > 1 GB)` `- Paging File % Usage` `  - Avg of 27% and max of 48%, so a lot above what's recommended (B.Ozar recommends "0" or "1")` `# PhysicalDisk - OK` `- Read Latency Analysis` `  - Averages for all drives are well under 100 milliseconds (B.Ozar recommendation)` `  - On Mon morning (4/20/2020) 4:21-10:53am, the C:\ drive spiked to 316 milliseconds`

```
    - This seems to be a one-
off
- Write Latency Analysis
  - Averages for all drives
are well under 100
milliseconds (B.Ozar
recommendation)
  - No spikes over 100
milliseconds

# Processor - OK

- % Processor Time
  - The average utilization
is well under 50%, but there
are night spikes to around
80% (most likely due to jobs)

# SQLServer:Buffer Manager -
OK

- Page life expectancy
  - While the average value
is 321,829 seconds, it made a
vertical drop to 15 on Sun
night (4/19/2020)
    * Probably caused by an
index rebuild or by the VM
  - Note: B.Ozar recommends
at least 180 seconds, while
sqlwatch recommends > 300

# SQLServer:General
Statistics - OK

- User Connections
  - Number of user
connections spike around noon
time
    * Max connections is 245

# SQLServer:Memory Manager -
OK

- Memory Grants Pending
  - All zeroes, as
recommended by B.Ozar

# SQLServer:Batch Statistics
- OK
```

			`` - Re-Compilations/sec - The ratio percentage of SQL Re-Compilations to SQL Compilations has an average of 0% - On Fri (4/17/2020) before 5:28pm, this spiked to 9% (sqlwatch recommends < 10%) - This appears to be a one off # System - OK - Processor Queue Length - It is on average below 10 threads per processor, which is acceptable ```

B. Update Compatibility Level with QTA

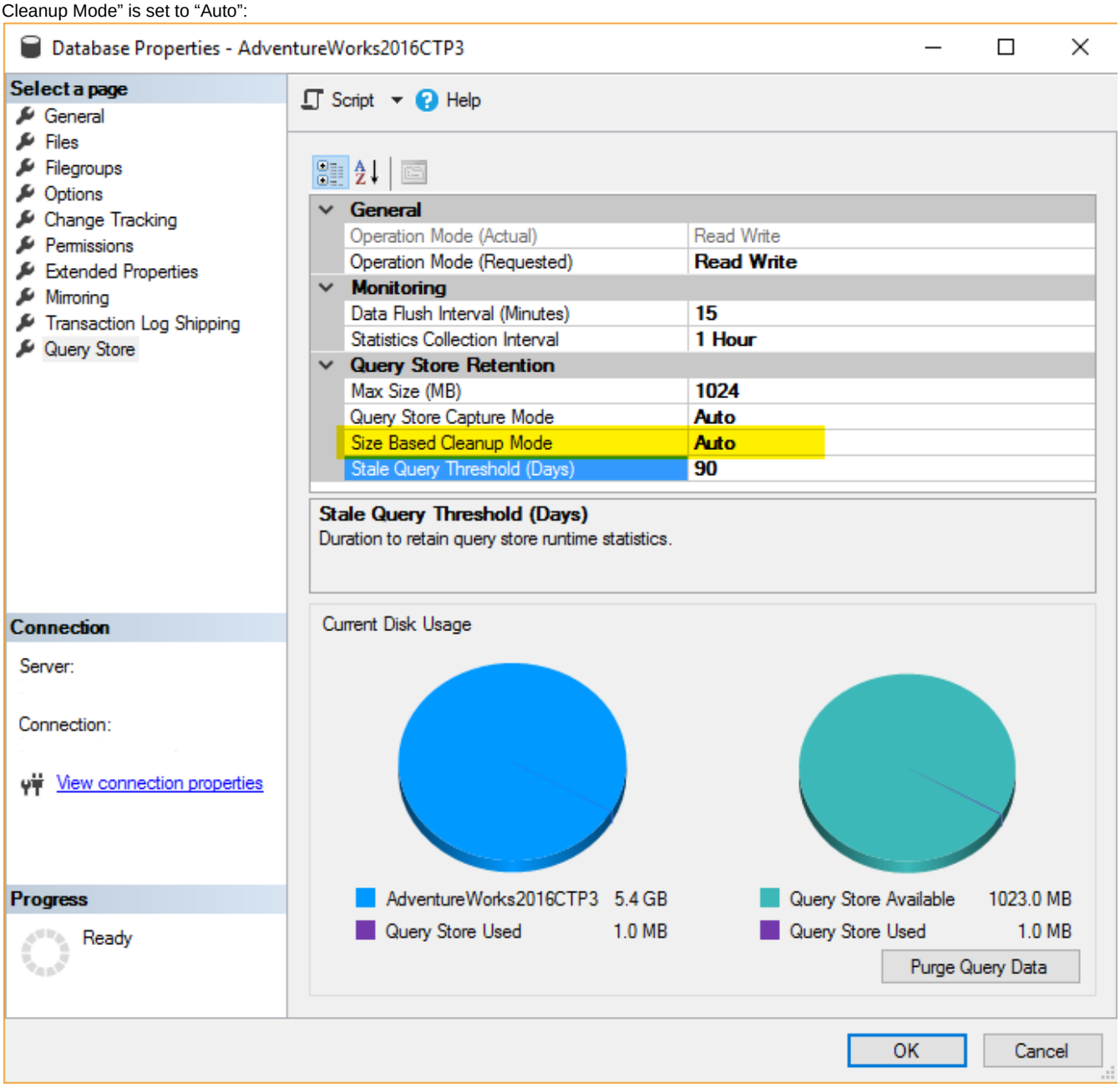
I. Creating baselines with old and new compatibility levels

	Task	Pending	Notes
1	☐ Start a “New Database Upgrade Session” by right-clicking the database in SSMS and going to “Tasks > Database Upgrade”	☐	Recommended Settings: • Use 7 days for the workload duration, unless otherwise noted by the customer • Select the highest target compatibility lv possible • Check the current plan cache size using a query like: ``` -- https://stevestedman.com/2012/08/tsql-to-determine-plan-cache-size/ select name, sum(pages_kb)/1024.0 MBUsed from sys.dm_os_memory_clerks where name = 'SQL PLans' group by name; ``` • If the cache size is is <= 1024MB, use the Recommended settings • If more, select Current and manually set a Max Size greater than the cache size. Copy the recommended settings for everything else
2	☐ After the specified workload duration has passed, check “Done with workload run” (doing so will update the DB compatibility automatically)		

II. Performance tune regressed queries

	Task	Pending	Notes
1	☐ Once the workload duration has passed for the new compatibility lv, check “Done with workload run” and tune any regressed queries	☐	Recommended Settings: • In the Analysis tab, select all tunable queries • In the Findings tab, only select queries with a positive % Change

Per this MS KB and article, the Query Store automatically keeps it data below 90% of the Max Size (set in step I1 above) if it’s “Size Based Cleanup Mode” is set to “Auto”:



Even if it falls behind and switches into read-only mode, this switch is only “temporary” and, per this MS KB, will switch back to read-write after enough space is cleared. Unless there are extenuating circumstances, it should be safe to leave on the Query Store, which is also enabled by default in Azure.

Things to explore for improvement:

- Using SQLWATCH or another performance reporting tool <https://sqlwatch.io/ >
- Perfmon recommendations (counters to collect and expected numbers) from “SQL Server Query Performance Tuning” book