

Pre-Upgrade Planning and Testing Checklist

The following checklist offers informative steps and tips on how to prepare your pre-upgraded SQL Server database for SQL Server 2016/2017.

Release Notes:

As a best practice, it's beneficial to review the release notes for the SQL Server upgrade you have chosen to update your system to.

- [SQL Server 2016](#)
- [SQL Server 2017](#)

Review Requirements:

Your upgrade will always perform better with up-to-date [hardware and software](#). Depending on the state of your hardware/software certain aspects of your upgrade method/process may be affected (i.e., new hardware could reduce licensing via less processors or database/server consolidation).

Review Current Environment:

Understanding the SQL Server components in use and clients connected to your environment is vital to upgrading. Research and review your current SQL Server environment.

UPDATING CLIENT PROVIDERS

Though updating your provider for clients isn't necessary when upgrading, you may find it beneficial to your new environment. Users on certain versions (i.e., SQL Server 2014 [12.x] or older) may have to update client providers for new features like [Always Encrypted \(DE\)](#), [Stretch Database](#), and [others](#).

Third-Party Compatibility:

As a safety precaution, it's important to determine the compatibility status of your third-party components (i.e., integrated backup).

BACKWARD COMPATIBILITY

On the topic of compatibility, Microsoft offers helpful information on how to properly review your SQL Server database engine backward compatibility.

The full article can be found [here](#).

Analyze Target Environment:

Just as you would review the requirements of your hardware/software for the upgrade, it's helpful to analyze target environments, making sure they meet all requirements. For instance, virtualization of SQL Server to a public and/or private cloud, or consolidating multiple instances into your new SQL Server instance.

Edition:

Deciding on an appropriate edition of SQL Server as your upgrade will help you direct the path you take toward upgrading. Before making any concrete decisions on a [specific edition](#), review the options, understand and verify the functionality, and make sure it will transfer over into your new environment.

Assurances:

To make sure your upgrade and data migration go as smoothly as possible, there's a couple check processes you can run for assurance purposes: Migration Assistance and System Configuration Checker.

RUN MIGRATION ASSISTANT

Any issues during the course of an upgrade could result in damaged or loss of data; as a best practice, be sure to run SQL Server's Data Migration Assistant. This tool will help you identify and assess any potential issues that could hinder your upgrade.

RUN CONFIGURATION CHECKER

The SQL Server System Checker (SCC) is another useful tool to assist your team in identifying blocking issues in the SQL Server setup program.

Memory-Optimized Tables:

If your SQL Server database instance holds any memory-optimized tables (MOTs) you'll want to schedule additional time for table conversion. Depending on the size of your MOTs and speed of the I/O subsystem. Ensure there is enough space on the disk before proceeding or the process may cause recovery to fail. Such an upgrade requires three operations for in-place and new installations (the first is not required for rolling upgrades):

1. Run database recovery with current/old on-disk configuration
2. Serialize data to disk for new on-disk configuration
3. Run database recovery on new configuration

Once you've prepared your database engine for deployment, you'll want to treat the upgrade as you would any other regular IT project; organize your team and begin developing/testing your upgrade strategy. Your team will determine and conduct the following activities:

Upgrade Method:

There are a number of ways your team can approach upgrade process. Primarily, users can perform an in-place upgrade, migrate to a new installation, or conduct a rolling upgrade (all of which are designed to minimize downtime/risk in the course of your upgrade). For more information on how to choose, review the documentation offered [here](#).

Create Rollback plan:

Just in case your upgrade doesn't turn out as expected, developing and executing a strong rollback strategy will help you restore your original environment.

Review Criteria:

Before transferring over other users, be sure that everything has gone as planned. Verify that every aspect of the upgrade has gone as planned and the new, upgraded environment is up and operational.

Test Your Plan:

Now that you've double-checked your methodology and developed rollback/criteria plans, you should now test the performance of your upgrade strategy. Using the [Microsoft SQL Server Distributed Replay Utility](#), multiple computers can replay trace data and simulate mission-critical workloads. By doing this on test servers before and after the upgrade, your team will be able to measure performance differences and analyze potential incompatibilities your upgrade may have.

For more information or assistance in upgrading your SQL Server environment to SQL Server 2016 or SQL Server 2017, please reach out to us at info@xtivia.com.