

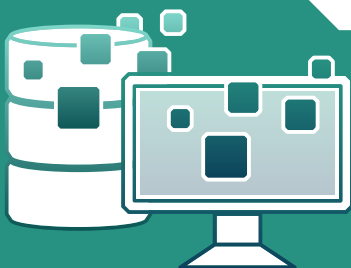


Making the move to Microsoft SQL Server 2017 and Microsoft Windows Server 2016 on a Dell EMC PowerEdge R740xd

Making the move to the latest database software can seem a daunting task, but it doesn't have to be difficult. By migrating your Microsoft® SQL Server® 2008 R2 instances onto a new Dell EMC™ PowerEdge™ R740xd server running Microsoft SQL Server 2017, your organization can secure the performance, security, and management benefits that a refresh can provide.

In the Principled Technologies datacenter, we set up an older infrastructure server (Microsoft SQL Server 2008 R2 and Microsoft Windows Server® 2012 R2 on a Dell EMC PowerEdge R720xd) and migrated a database from it to a new solution: Microsoft SQL Server 2017 and Microsoft Windows Server 2016 on a 14th generation Dell EMC PowerEdge R740xd server.

In this migration guide, we use our expertise and experience to help you research and plan for your migration, share best practices and recommendations, and give you step-by-step guidelines to make your move to the new solution as smooth as possible.



Dell EMC PowerEdge R740xd

Contents

Why migrate?	3
Better performance and consolidation potential	3
New features bring new possibilities	3
Maintain support and stay compliant	3
We show you how	4
Before the move	4
Installation considerations	4
Hardware and software requirements	4
A look at our new solution	5
Server core considerations	6
Working in a multi-version environment	6
Evaluating the source environment	6
Using Data Migration Assistant	6
Installing Windows Server 2016 and Microsoft SQL Server 2017	8
Configuring the Dell EMC PowerEdge R740xd internal storage	8
Installing Microsoft Windows Server 2016	9
Installing Microsoft SQL Server 2017	9
Validating the installation	13
Making the move	14
Performing the backup of the SQL Server 2008 R2 database	14
Restoring the database to the SQL Server 2017	16
After the move	18
Post-migration considerations	18
Moving Windows and SQL Server logins	18
SQL Server Agent jobs	20
Other external dependencies	20
Summing it all up	21
System configuration information	22
Installing the SQL Server environment	24
Backing up and restoring the database	25
Post-installation tasks	26

Why migrate?

If you're happy with how your datacenter is running, you may ask yourself: Why migrate? What's in it for me? Every investment costs time, money, and effort from your staff, so it's understandable that you'd like to make sure a refresh is worth it. We're here to assure you that it usually is, for the following reasons:

Better performance and consolidation potential

New versions of software and hardware incorporate the industry's latest technological advances, so they often outperform their older counterparts. In our own datacenter, we compared the database and business analytics performance of the legacy Microsoft + Dell EMC solution to Microsoft SQL Server 2017 on the new Dell EMC PowerEdge R740xd, and found that newer was better. The new solution supported nearly seven times as many customer orders and answered business analytics queries in 90 percent less time, helping you get data insights faster.¹

With this significantly improved performance, you could potentially consolidate the machines you're already running and save on maintenance and operating costs over the long term by running fewer servers.

New features bring new possibilities

New software and hardware bring new features that can improve security, maintenance, and flexibility for your database administrators. See our feature sections on page 5 to learn more about the new features that Microsoft SQL Server 2017 and Windows Server 2016 on a Dell EMC PowerEdge R740xd can deliver for your organization.

Maintain support and stay compliant

Software vendors don't continue to support outdated software versions forever. If you want your databases to remain secure and be compliant with industry regulations (e.g., HIPAA, GDPR) you must upgrade to supported software versions. According to Microsoft, extended support for SQL Server 2008 and 2008 R2 ends on July 9, 2019. This means that after that date, Microsoft will stop releasing vital security updates.²

Migrating from a legacy database solution isn't as hard as it looks. In the following sections, we provide step-by-step instructions to help you plan and enact a successful migration to Microsoft SQL Server 2017 and Windows Server 2016 on the Dell EMC PowerEdge R740xd.

1 Read our full reports at <http://facts.pt/ctNcYN> and <http://facts.pt/rviwgy>.

2 "Prepare for SQL Server 2008 end of support," accessed June 14, 2018, <https://www.microsoft.com/en-us/sql-server/sql-server-2008>.

We show you how

Before the move

We assume that you have an existing installation of Microsoft SQL Server 2008 R2, complete with the necessary infrastructure and support, and understanding of the installation and surrounding infrastructure. In this guide, we'll show you how to:

- build the new Microsoft SQL Server 2017 environment
- create a database backup on your existing SQL Server 2008 R2 instance
- transfer the database from the existing environment to the Dell EMC PowerEdge R740xd
- upgrade the database to SQL Server 2017
- transfer logins and agent jobs

Please recognize that the method we outline in this guide is not the only way to migrate to SQL Server 2017—other methods include detach/attach, backup/restore, the Data Migration Assistant, and the Copy Database Wizard. We will show you how to use native SQL Server backup files to transport user database files to the new server. For considerations we don't cover in this guide, see msdn.microsoft.com/en-us/library/bb677622.aspx.

To complete the migration, you'll need to plan a maintenance period where your database is in a read-only state, meaning users can view the database but may not change it. This step is required so that the database will retain any modifications made during migration. The length of read-only time depends on how long it takes you to create a database backup, copy the database backup to the new Microsoft SQL Server 2017 instance, and to restore the database.

Installation considerations

Hardware and software requirements

The following is a summary of the basic hardware and software requirements for installing Microsoft SQL Server 2017:

- .NET™ Framework 3.5 SP1 and 4.6 for 2017
- Windows PowerShell® 2.0
- Network software (Shared memory, Named Pipes, TCP/IP or VIA)
 - Shared memory and VIA are not supported on failover clusters, and VIA is now deprecated and will be removed in a future version of SQL Server.
- 6 GB of drive space
- RAM
 - Minimum: 1 GB of RAM (512 MB for Express Editions)
 - Recommended: 4 GB or more depending on database size (1 GB for Express Editions)
- Processor
 - Minimum: 1.4 GHz
 - Recommended: 2.0 GHz or faster

For a complete list of hardware and software requirements, and a list of supported operating systems for each SQL Server edition, see <https://docs.microsoft.com/en-us/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server?view=sql-server-2017>.

A look at our new solution

For this guide, we migrated to a new solution featuring Microsoft SQL Server 2017 Standard, Microsoft Windows Server 2016, and a 14th generation Dell EMC PowerEdge R740xd server.

Microsoft SQL Server 2017

If you're updating your SQL Server editions from 2008 to 2017, you'll find a host of new features to benefit your business. The SQL Server 2017 features could:

- Speed up online transactional processing with in-memory optimization, which takes tables of transaction data from the system memory instead of pulling them from disk storage
- Decrease the load on your drives with Columnstore Indexes, which store and process data in columns instead of rows
- Improve I/O throughput with the buffer pool extension, which extends memory buffer to SSDs

In addition to these new performance features, SQL Server 2017 also offers basic availability groups (to support failover environments) and additional security and management features. It has expanded its cloud-readiness with services like backup and disaster recovery in Microsoft Azure®, and added several business analytics features that help businesses gain more insight into their data.

Dell EMC PowerEdge R740xd

The 14th generation Dell EMC PowerEdge R740xd offers strong database performance with a variety of storage configuration options. Although in our performance tests we used a combination of SAS HDDs and SATA SSDs, the PowerEdge R740xd can also support low-latency NVMe drives and new NVDIMM technologies. The 14th generation server also supports up to 3 TB of RAM—double that of the previous-generation R730xd. The PowerEdge R740xd features processors from the Intel® Xeon® Scalable processor family.

Microsoft Windows Server 2016

Updating your Windows Server edition from 2012 R2 to 2016 also opens up numerous security, compute, networking, storage, and cloud-ready features:

- Host more VMs with increased memory and logical processor support, on both physical and virtual machines
- Take advantage of containers (which make applications more portable and can use fewer server resources) with full support for Windows Server and Hyper-V™ containers
- Protect your VMs from compromised fabric with shielded VMs
- Minimize downtime during reboots and upgrades with Cluster OS Rolling Upgrade, which moves VM clusters between hosts to enable rolling upgrades
- Access the high availability and scalability of software-defined storage with Storage Spaces Direct

Server core considerations

You can install Microsoft SQL Server 2017 on most Microsoft Windows Server Core versions of Windows Server 2012 and Windows Server 2016. Some features cannot be installed on Microsoft Windows Server Core versions, so you must install them on a different machine. For a list of which features are supported, see msdn.microsoft.com/en-us/library/hh231669.aspx.

Working in a multi-version environment

It's possible to install distinct versions of SQL Server (e.g., SQL Server 2016 and SQL Server 2017) on the same machine, but certain considerations apply. For a detailed list of these considerations, see msdn.microsoft.com/en-us/library/ms143694.aspx.

Evaluating the source environment

As with any migration or consolidation, planning is key. There are specific details related to each server targeted for migration or upgrade, including the maintenance window for migrating the databases to their new environment, metadata, affected users, and configuration tasks necessary to assimilate the databases into the consolidated environment. Before migration, consider the following, and develop documentation, scripts, or procedures to deal with each item:

- Windows Server version and patch level
- SQL Server version and patch level
- Number of logins on this SQL Server instance, and what type of logins these are (Windows or SQL)
- Current backup strategy and schedule for the databases on this server
- Replication details for this SQL instance, if any
- Detailed information regarding permissions and roles
- SQL Server Agent jobs on this SQL Server

After moving the databases to their new SQL Server instance, make sure that any system or application using the database has updated connection information. This includes logins, permissions, applications, SQL Server Agent jobs, third-party backup products, and so on.

Using Data Migration Assistant

The SQL Server 2017 Data Migration Assistant (DMA) can be a major aid in migration research. You can also use Data Migration Assistant to migrate your database instead of doing so manually, if you desire. This utility, which you can download from <https://www.microsoft.com/en-us/download/details.aspx?id=53595>, scans prior versions of SQL Server databases and SQL Server components for compatibility issues, features, and T-SQL syntax the newer version may not support. The Data Migration Assistant interface lets you view or save reports that list warnings and errors.

The Data Migration Assistant works on machines running Windows® 7, Windows 8, Windows 10, or Windows Server 2012, Windows Server 2012 R2, or Windows Server 2016. The Microsoft .NET framework (4.5.2 or above) is also required. For more information about Data Migration Assistant, see <https://docs.microsoft.com/en-us/sql/dma/dma-overview?view=sql-server-2017>.

To install the Microsoft SQL Server 2017 Data Migration Assistant, perform the following steps (for greater detail, see [Installing the SQL Server environment](#)):

1. To install DMA, download the latest version of the tool from the Microsoft Download Center, and run the DataMigrationAssistant.msi file.
2. Following the prompts, install DMA with defaults.
3. Select Start→All Programs→Microsoft Data Migration Assistant→Microsoft Data Migration Assistant.
4. To create a new project, click the plus sign (+ New).
5. Select Migration→Assessment, name the project, and change Target server type to SQL Server.
6. In Options, check Compatibility Issues, and click Next.
7. In Select sources, enter the older SQL Server computer name, put in your authentication credentials, and click Connect.
8. At the prompt, select all the databases you wish to migrate over, and click Add.
9. After you've added your servers, click Start Assessment.
10. Enter the older SQL Server computer name, and select the features you want the Upgrade Advisor to analyze (see Figure 1). Alternately, the Upgrade Advisor can remotely scan the older SQL Server machine and detect the components running on the server. To do this, click Detect.

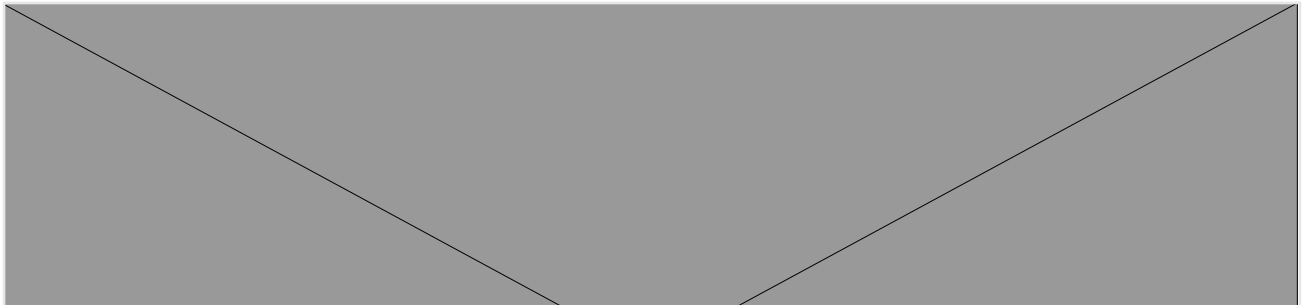


Figure 1: Upgrade Advisor component selection

11. Choose the instance, provide proper authentication to the older SQL Server, and select the database you want to analyze (see Figure 2). Or, if you've captured SQL trace files using SQL Profiler or have a particular T-SQL batch to check, input SQL script files and trace files at this time.

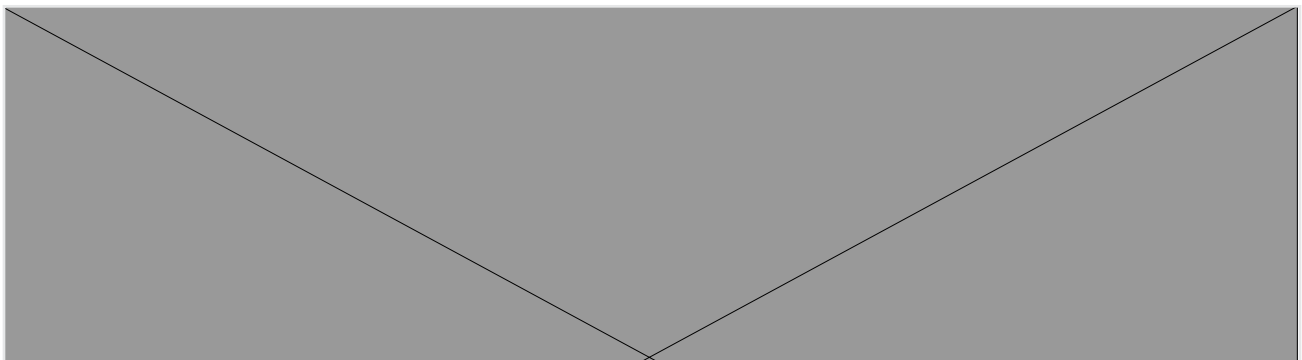


Figure 2: Upgrade Advisor sources selection

- Following the analysis of the older SQL Server instance, view the Upgrade Advisor report. Each warning or error will have associated information to solve any issues (see Figure 3).

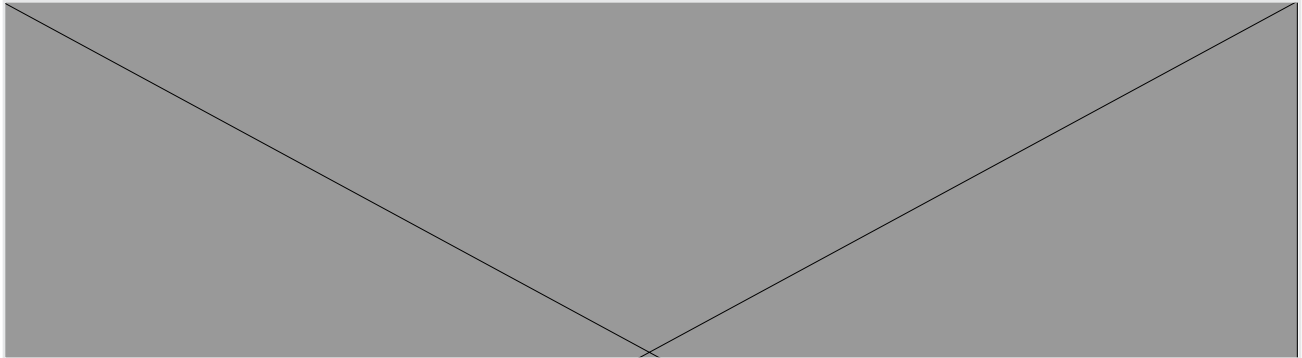


Figure 3: Sample Upgrade Advisor output report

Gathering baseline performance data

During the research and planning phase for this migration, use Windows Performance Monitor, SQL Server Profiler, SQL Server Distributed Replay, and other tools to gather data on the typical query load and performance statistics of the databases potentially moving to the new environment. This serves two purposes: first, it provides an opportunity to identify potential problems before moving to the new solution; second, the information can help to analyze resource allocation needs. For more information on SQL Server Profiler and SQL Server Distributed Replay, see msdn.microsoft.com/en-us/library/ff878183.aspx.

Installing Windows Server 2016 and Microsoft SQL Server 2017

After the prerequisite infrastructure is in place and you've planned your migration, you're ready to set up the new Microsoft SQL Server 2017 environment on the Dell EMC PowerEdge R740xd. For more detailed steps, see [Installing the SQL Server environment](#). Dell EMC and Microsoft also have multiple management and deployment options available to automate the server deployment process, such as OpenManage Essentials and OpenManage Enterprise with integrations for Microsoft System Center. Although in this report we used a manual sequence in order to best describe the required operations, using OpenManage could further simplify this process. Learn more about OpenManage Systems Management [here](#).

Configuring the Dell EMC PowerEdge R740xd internal storage

For our testing purposes, we used only internal storage on the Dell EMC PowerEdge R740xd. We created three RAID groups: a RAID1 pair of HDDs for OS, a RAID1 pair of SSDs for logs, and a 10-SSD RAID10 for data. If your environment uses external storage, you may need further planning before you attempt this migration.

- To enter the PERC H740P Configuration Utility during POST, press Ctrl+R at the prompt.
- Select the controller, and press F2. Select Create VD.
- Specify the RAID level, select the desired physical disks, choose the appropriate cache settings, and press Apply.
- Repeat for each RAID group.
- Exit the configuration utility, and reboot the server.

Installing Microsoft Windows Server 2016

After configuring the internal storage on the Dell EMC PowerEdge R740xd, install the operating system. We installed Windows Server 2016 with default options. The Server Manager feature lets you add Roles and Features to the installation after the initial configuration. To install the operating system:

1. Boot the Dell EMC PowerEdge R740xd into the Microsoft Windows Server 2016 installation menu.
2. Select the appropriate language, time and currency format, and keyboard layout. Enter the product key, and choose Windows Server 2016 Datacenter (Server with a GUI).
3. Accept the license terms, and install the OS to the appropriate disk.
4. After the installation has finished, configure Microsoft Windows Server 2016 as your specific environment requires with any security policies, Active Directory requirements, and so on.

Installing Microsoft SQL Server 2017

Now that your operating system is running, you may install the new database software. To follow best practices, deploy SQL Server 2017 on machines that are part of an Active Directory® domain. If you deploy SQL Server in a domain, do not make the SQL Server service domain accounts members of the Domain Administrators group. Grant only the necessary rights on the local server to the SQL Server service account as part of the pre-installation planning. The SQL Server installation software creates the local groups it needs for security purposes. For more information on SQL Server service accounts, go to msdn.microsoft.com/en-us/library/cc281953.aspx.

1. Using Server Manager, add the .NET Framework 4.5 feature, and if necessary, reboot.
2. Attach the Microsoft SQL Server 2017 installation media to the server, and start the SQL Server 2017 setup.
3. Select Installation, and click New SQL Server stand-alone installation or add features to an existing installation (Figure 4).



Figure 4: SQL Server 2017 Installation Center

4. Follow the installation wizard, confirming installation type and providing licensing details.
5. At the Feature Selection screen (Figure 5), select the features required by your organization. We selected

Database Engine Services, Full-Text and Semantic Extractions for Search, Client Tools Connectivity, Client Tool Backwards Compatibility, Management Tools – Basic, and Management Tools – Complete. Specify a directory, and click Next.

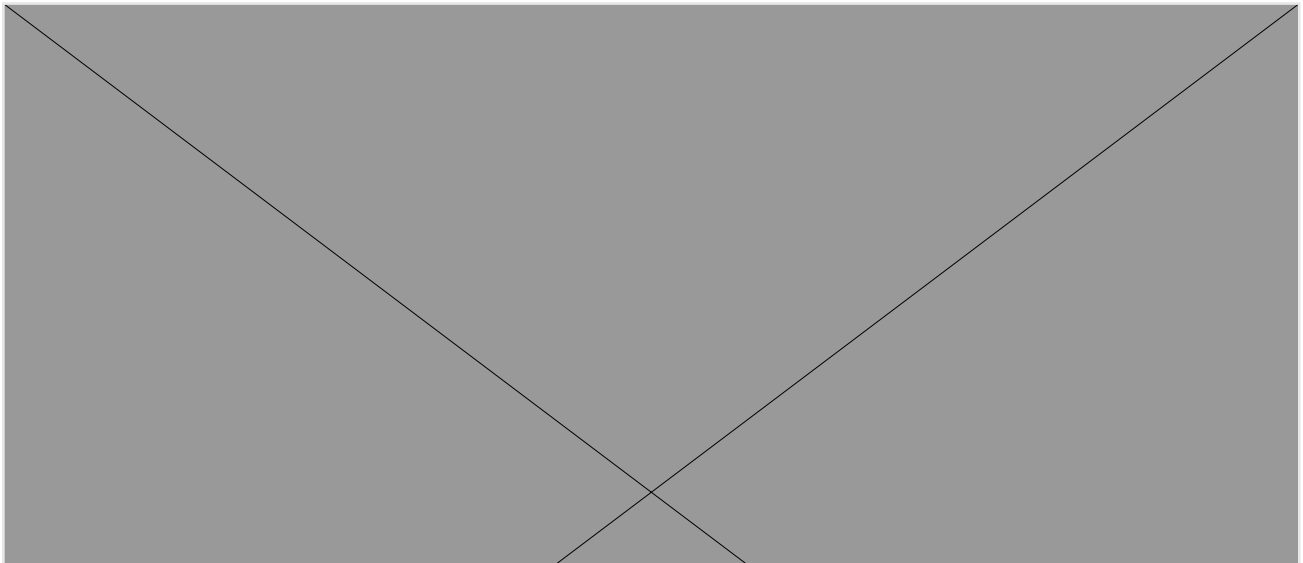


Figure 5: SQL Server 2017 Feature Selection screen

6. Choose Default instance or Named instance, depending on the needs of your organization. Click Next.
7. On the Server Configuration screen (Figure 6), choose the accounts to be used for each SQL Server component. We left the default accounts. If you have specific domain accounts to be used for SQL Server service accounts, adjust them on this screen. Click Next.
8. On the Database Engine Configuration screen (Figure 7), address the following tabs:

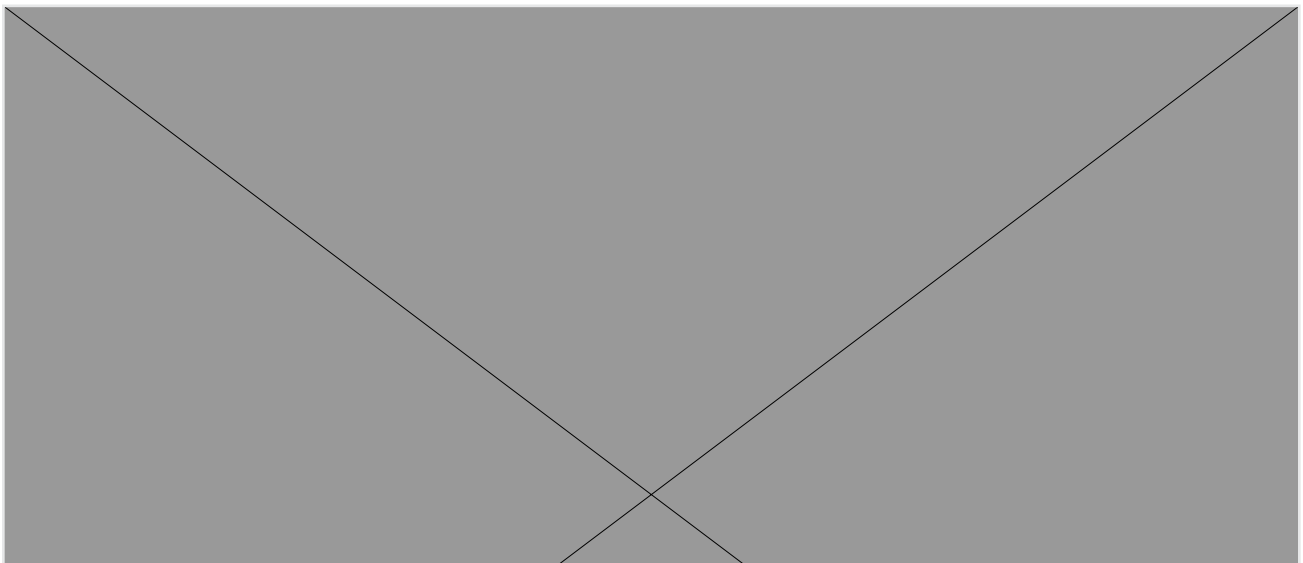


Figure 6: SQL Server 2017 Server Configuration screen

- a. Server Configuration: Select Windows Authentication Mode or Mixed Mode. Add any necessary users, and enter a password for the sa account if using Mixed Mode. We selected Mixed Mode and added the current user (Windows Local Administrator).
- b. Data Directories: Enter the default directories where the SQL Server installation will attempt to place its user database and log files upon creation.

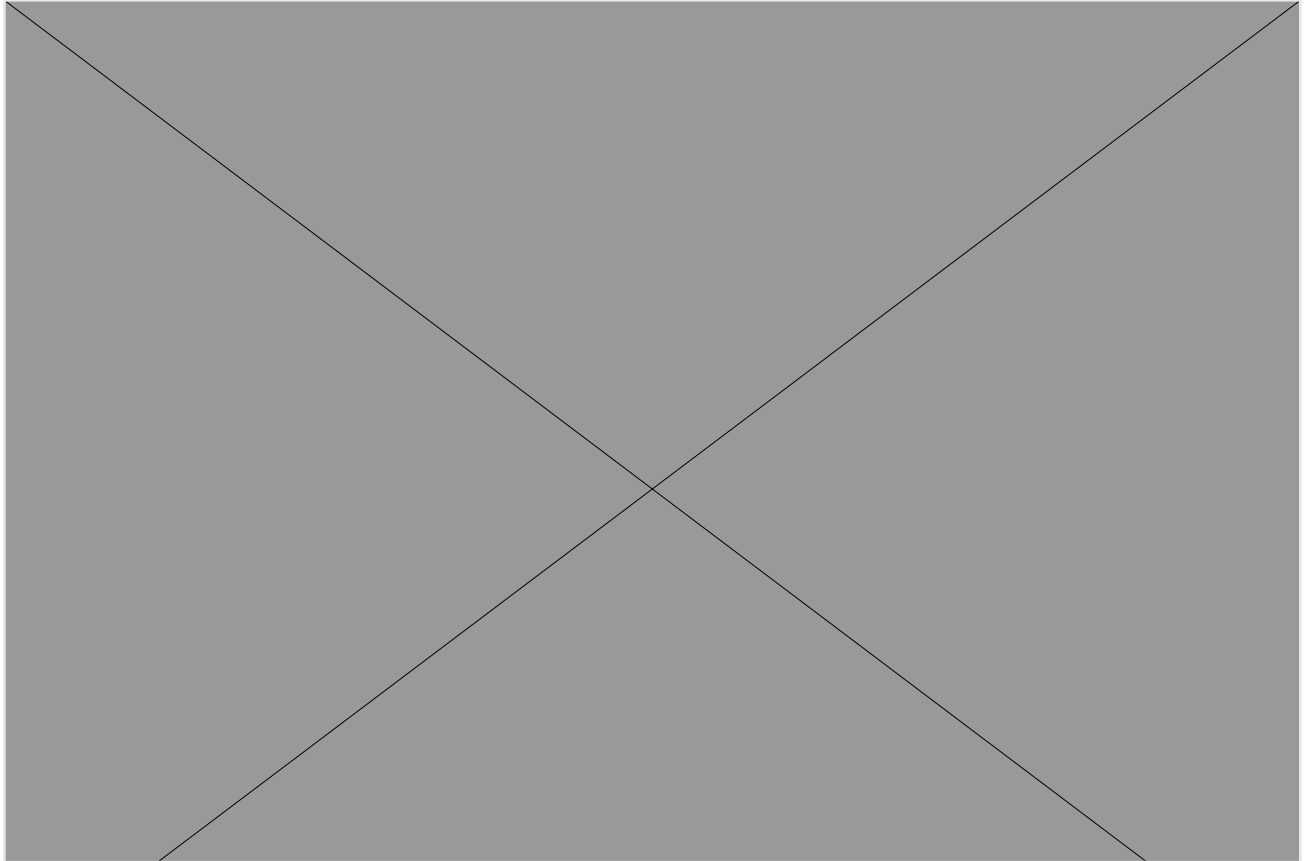


Figure 7: SQL Server 2017 Database Engine Configuration screen

- c. TempDB: Enter the number of files, initial size, and default directories for both the TempDB data and TempDB log files, and set an appropriate Autogrowth value. For our example, we used 8 TempDB data files with an initial size of 50GB and 10% autogrowth, and an initial size of 20GB for the TempDB log file.
 - d. FILESTREAM: If appropriate for your organization, enable FILESTREAM. We left FILESTREAM disabled.
9. Click Next.
 10. Let the Feature Configuration Rules check complete, and click Next.
 11. At the Ready to Install screen (Figure 8), review the summary info, and click Install.
 12. After the installation completes, click Close.

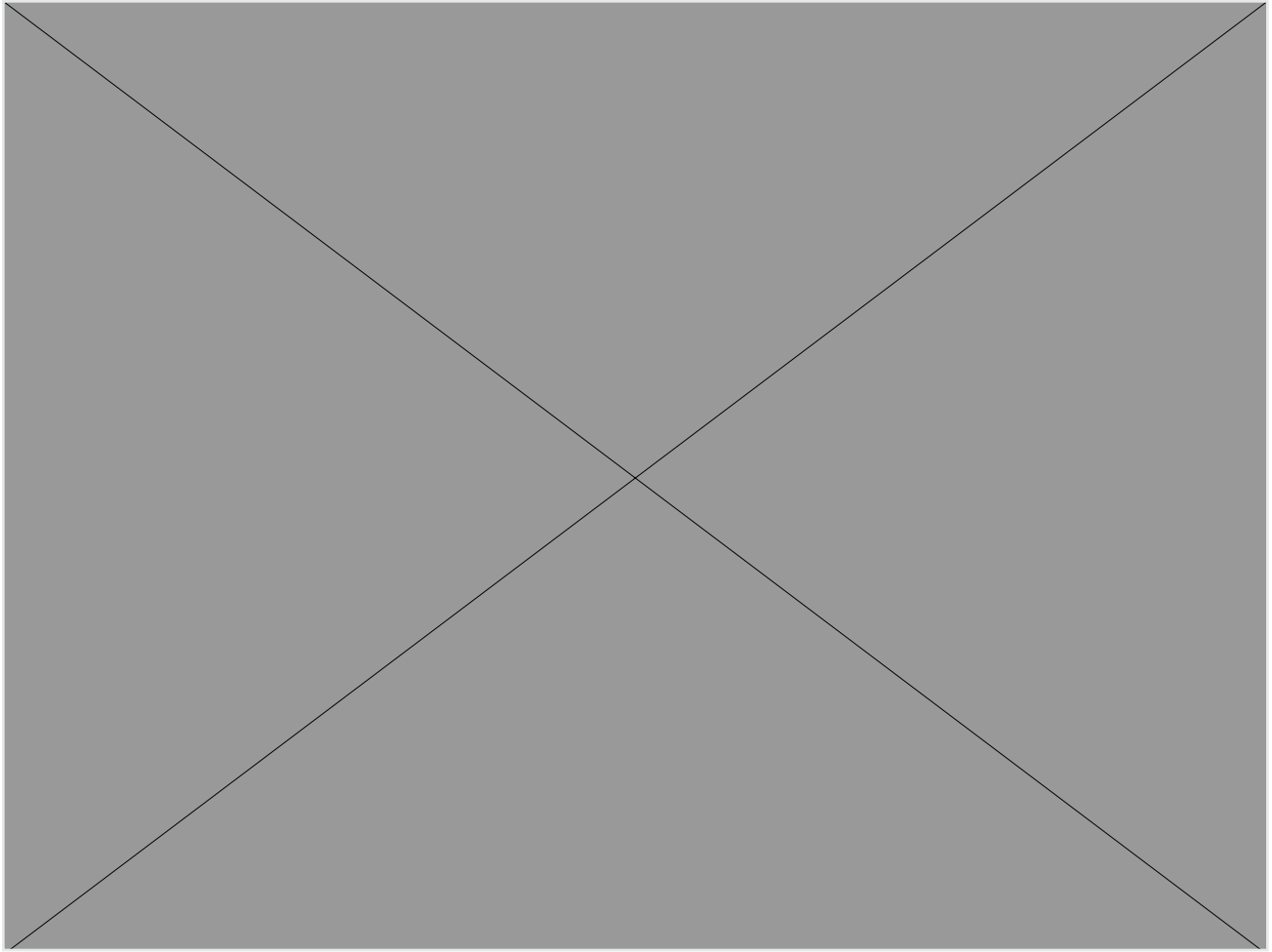


Figure 8: SQL Server 2017 Ready to Install screen

After completing the Microsoft SQL Server 2017 installation, complete any additional steps required to match your SQL Server 2008 R2 environment. We will outline the transfer of SQL Server logins and agent jobs in the [After the move](#) section.

Validating the installation

After completing the migration, the SQL Server features discovery report creates a list of all SQL Server 2017 products and features that are on the server. In this section, we outline how to create a basic report. For more detailed information, see msdn.microsoft.com/en-us/library/bb510455.aspx.

1. From the installation media, launch the SQL Server Installation Center.
2. Click Tools→Installed SQL Server features discovery report.
3. View the report in a Web browser, and verify that it matches the setup (Figure 9).

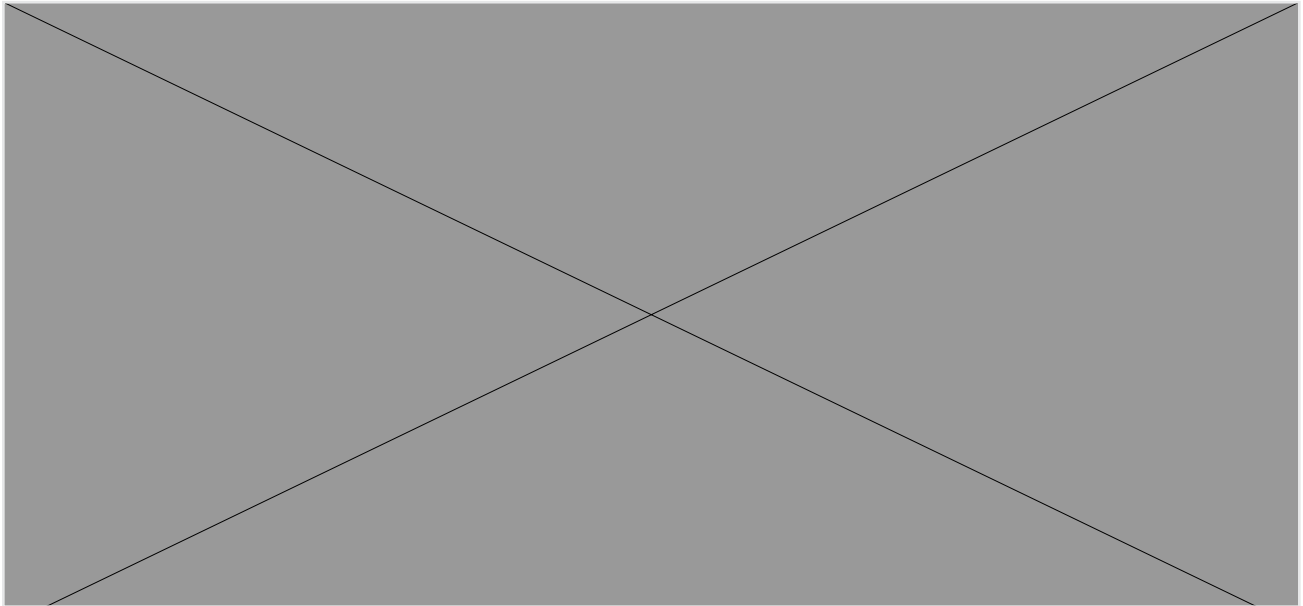


Figure 9: SQL Server 2017 Ready to Install screen

Making the move

In this section, we provide an overview of the processes involved in migrating a database from SQL Server 2008 R2 to a SQL Server 2017 environment on the Dell EMC PowerEdge R740xd. We performed all administration for the SQL Server 2008 R2 and SQL Server 2017 installations using SQL Server Management Studio, the main administration interface for those versions. For detailed steps on backing up and restoring the database, see [Backing up and restoring the database](#).

Various methods exist for migrating databases between instances and versions of SQL Server, including detach/attach, backup/restore, the Data Migration Assistant, and the Copy Database wizard. We chose the backup/restore method because most businesses will have a backup routine in place, and administrators will be familiar with the process.

Performing the backup of the SQL Server 2008 R2 database

Using the SQL Server 2008 R2 server, take the following steps to back up the database. Backup times will vary based on your specific database(s). To minimize downtime, do any special planning in advance.

1. To keep users from issuing updates during the migration process, set the database to read-only mode or set the access property to `SINGLE_USER` (Figure 10). Both choices immediately sever all user connections.

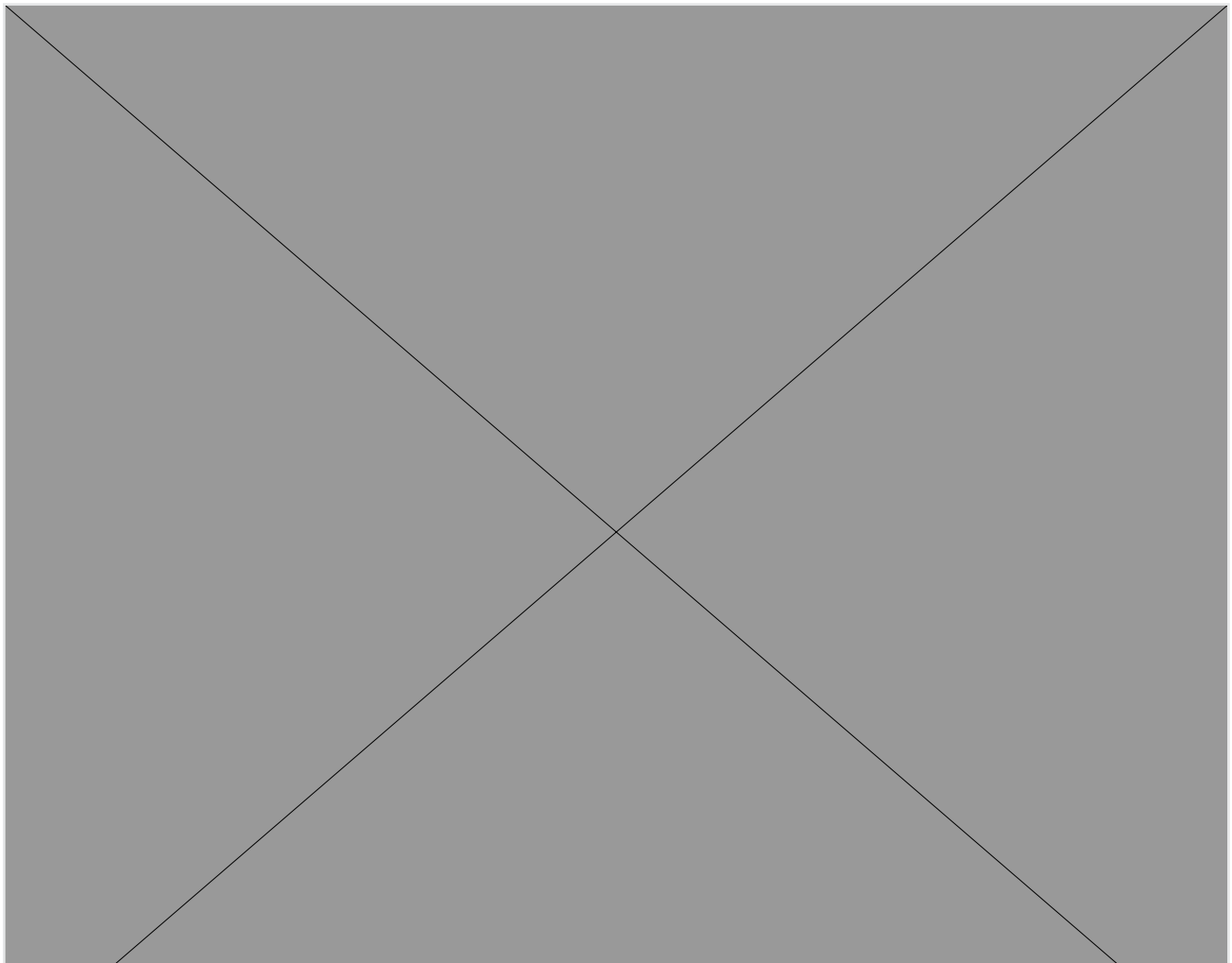


Figure 10: Setting the database to read-only or `SINGLE_USER`

2. Perform a full backup of the SQL Server 2008 R2 database (Figure 11).

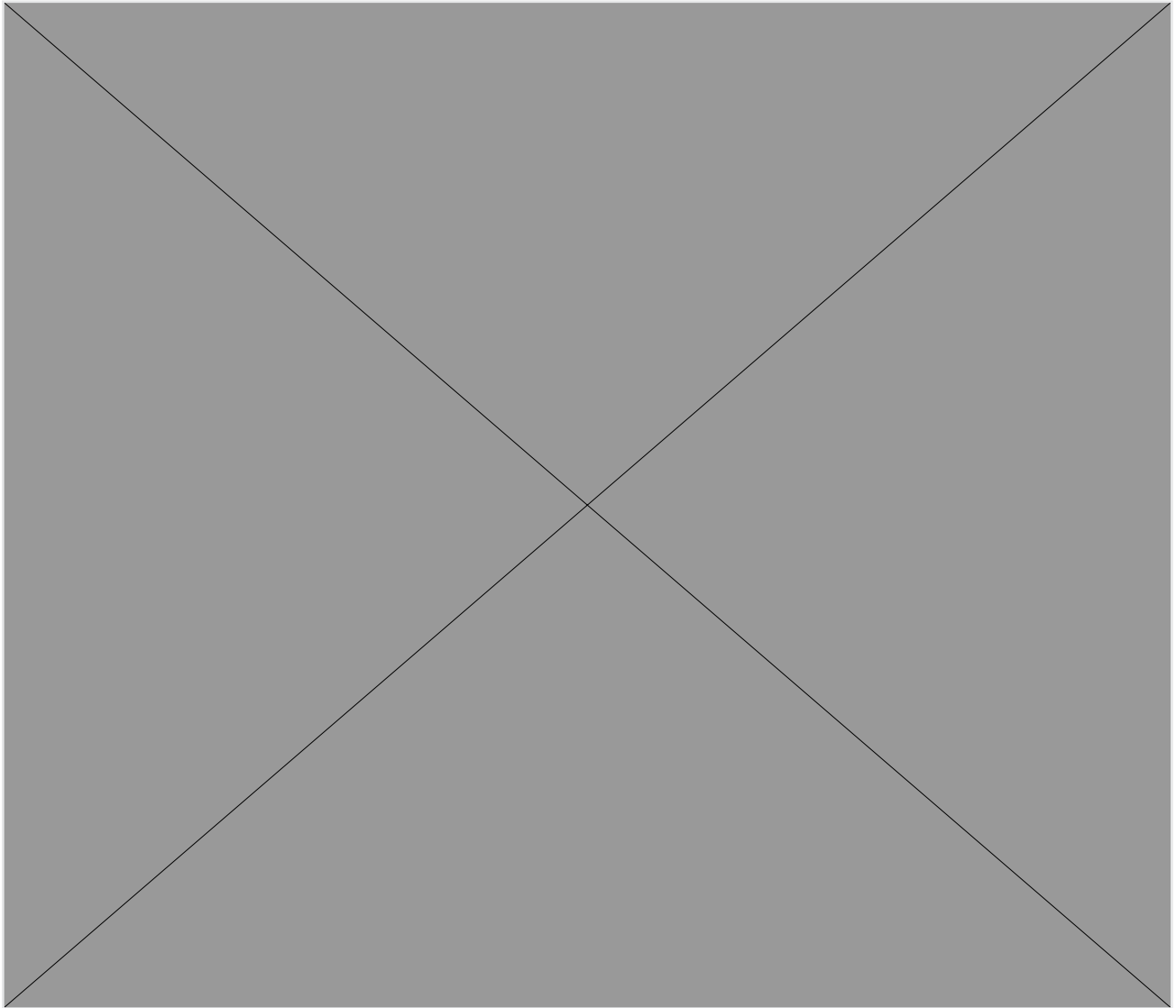


Figure 11: Backing up the SQL Server 2008 R2 database

If the time to complete a full backup is too long, take the full backup at a previously scheduled time, and at this point of the migration, take only a differential or transaction log backup. If the transaction log size is small relative to the data file size, this approach can decrease migration time significantly. If you choose this option, make sure to keep the backup chain intact.

Restoring the database to the SQL Server 2017

On the SQL Server 2017, take the following steps:

1. Open SQL Server Management Studio, connect to the SQL Server 2017 instance, and restore the database (Figure 12).

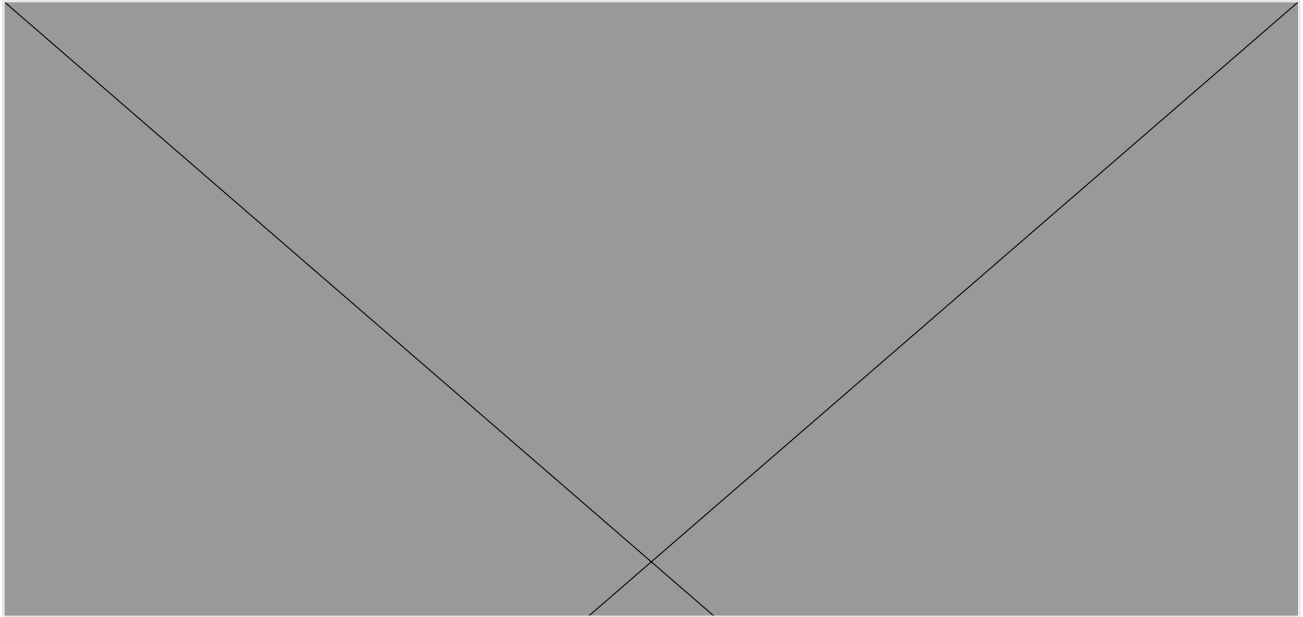


Figure 12: Restoring the SQL Server 2008 R2 database to the SQL Server 2017 environment on the Dell EMC PowerEdge R740xd

At this step, give the new database the same name as the SQL Server 2008 R2 database. Changing the name could cause applications that refer to the database by name to crash or not connect.

2. In SQL Server Management Studio, reset the database access property to multi-user, and set the compatibility level to SQL 2017 (level 140). See Figure 13.

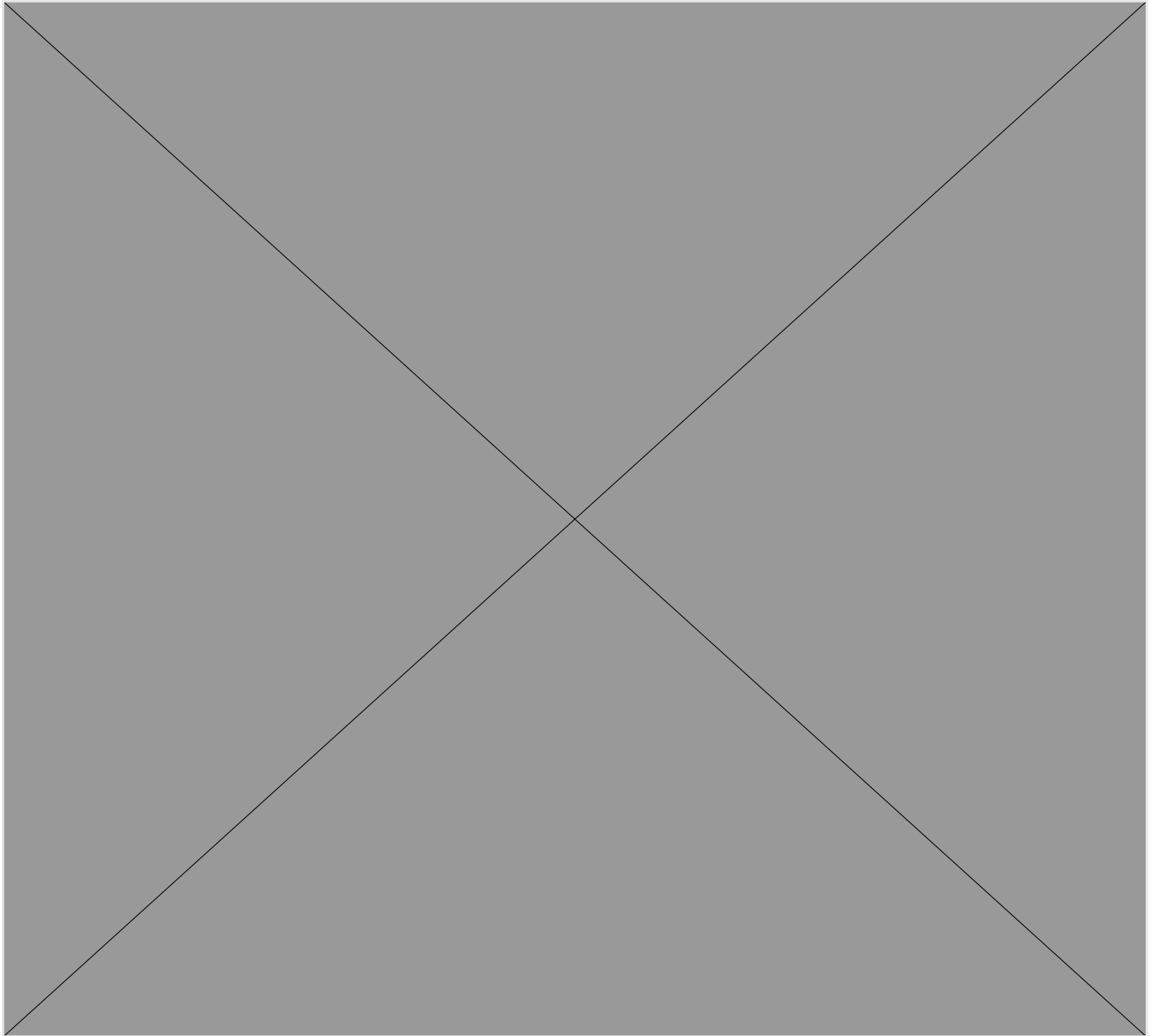


Figure 13: Setting the updated database back to read-write or MULTI_USER

3. Modify application connection strings or change Domain Name System (DNS) pointers that reference the migrated database.

After the move

Post-migration considerations

After the migration, you still have some tasks to complete. The specific list of tasks you must complete will depend on your pre-migration research and planning. In this section, we briefly discuss moving Windows and SQL Server logins and transferring SQL Server Agent jobs. See [Post-installation tasks](#) for detailed steps.

Moving Windows and SQL Server logins

As in previous versions of SQL Server, there are two methods of authenticating to SQL Server 2017: Windows logins and SQL Server logins. Windows logins are created and administered at either the operating system level or the Active Directory domain level, while SQL Server logins are created and managed from within SQL Server. The processes for extracting login information and creating the transferred login entities on the migration server are similar for both Windows-authenticated and SQL Server-authenticated logins. Take extra time and diligence, however, to ensure a smooth migration for SQL Server logins.

The following steps detail how to script both Windows-authenticated logins and SQL Server-authenticated logins to a query window and recreate them on the new SQL Server 2017 instance.

To transfer logins, take the following steps on the SQL Server 2017 instance. Note that steps 1 through 6 apply to both Windows logins and SQL Server logins, but step 7 refers only to SQL Server logins.

1. Log into the SQL Server Management Studio.
2. In SQL Server Management Studio and in the Object Explorer pane, connect to both the SQL Server 2008 R2 server and the SQL Server 2017 server. Be sure to have the Object Explorer Details tab open (View→Object Explorer Details).
3. Expand the tree view of the SQL Server 2008 R2 server, browse to the Security tab, and click the Logins node (Figure 14). The Object Explorer Details will show a list of all logins on the SQL Server 2008 R2 server.



Figure 14: SQL Server 2008 R2 logins

4. If necessary, use the sorting and filtering options in the Object Explorer Details tab, and take note of the logins you intend to migrate. Click to select them.
5. Right-click the selected logins, and choose Script Login As→Create To→New Query Editor Window. To change the connection of this new query window to the new SQL Server 2017 virtual server, right-click and select Change Connection (Figure 15).

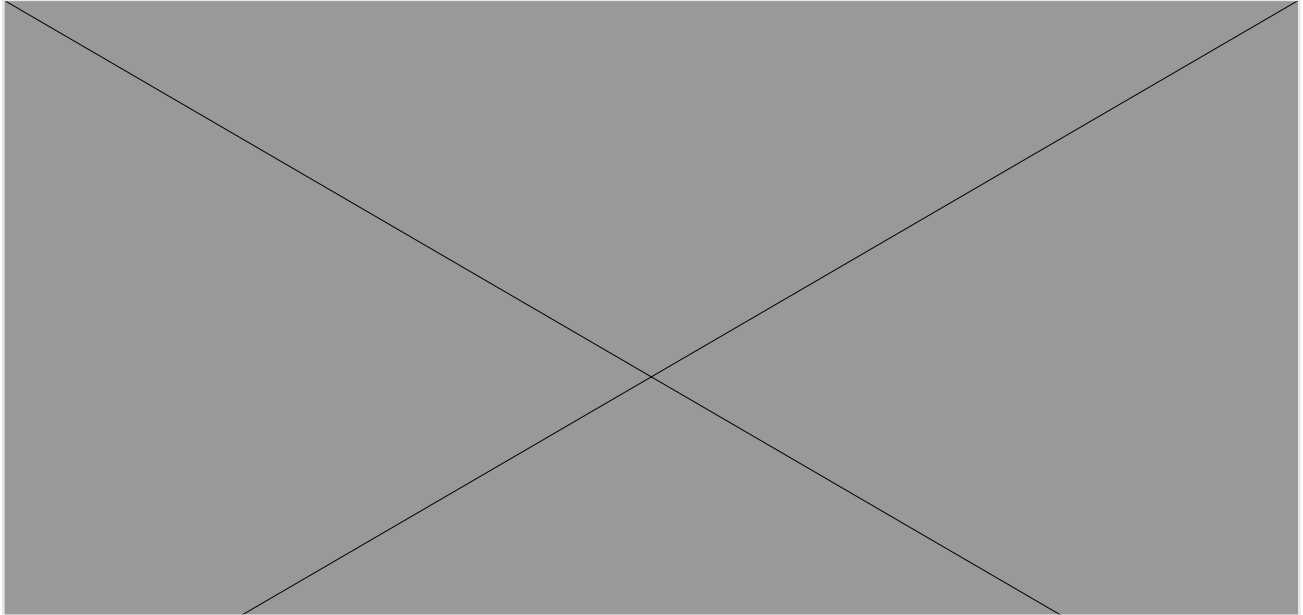


Figure 15: Changing the query editor connection to SQL Server 2017

6. To create the logins, execute the script on the SQL Server 2017 server. If transferring Windows-based logins, the process is complete. If transferring SQL Server logins, continue to step 7.
7. The script executed in step 6 created the login, marked it as disabled, and assigned it a random password. The script, however, did not map the login to the database user. To avoid having a database user that is orphaned from a login, use the `sp_change_users_login` stored procedure to view orphaned users and to map a user to a login. If using this method, your applications may need to account for new password reset or generation.

SQL Server Agent jobs

Most organizations schedule SQL Server Agent jobs that run against their databases, such as backups, index rebuilds, and other maintenance items. These jobs must be migrated to the new SQL Server 2017 instance. Log into the server, open SQL Server Management Studio, and connect to the older SQL Server 2008 R2 server. Locate the jobs to migrate in SQL Server Management Studio (on the SQL Server 2008 R2 server) under the SQL Server Agent, right-click them, and choose Script Job As→Create To→New Query Editor Window (Figure 16). Connect to the SQL Server 2017 instance, and run the resulting script in a query window. Verify the job was created successfully in SQL Server Management Studio.

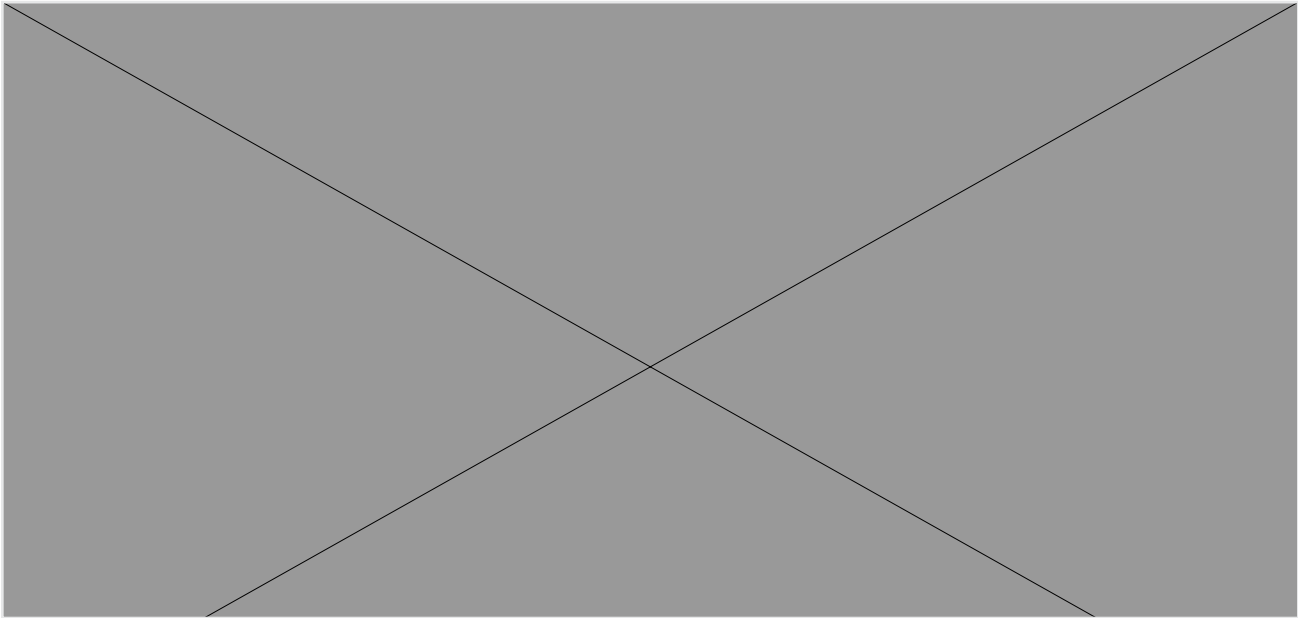


Figure 16: Scripting SQL Agent jobs in SQL Server Management Studio

Other external dependencies

Your pre-migration research may have yielded a list of items to implement, including references to local or remote file shares, database mail or SQL Mail configurations, stored procedures in system databases, and linked servers. Address these external dependencies to ensure a smooth finish to the migration.

Summing it all up

Preparing for your business's future means keeping your datacenter hardware and software up to date. If you've been running SQL Server 2008 or 2008 R2, the time to move to SQL Server 2017 is now. Extended support for SQL Server 2008 runs out in July 2019, which means that Microsoft will no longer release security updates, leaving your organization to face potential compliance issues. By moving to the latest SQL Server release, you can benefit from the new performance-enhancing features, the ability to choose either Windows or Linux operating systems, improved security features, and more. As we've demonstrated in this guide, making the move to SQL Server 2017 is straightforward and requires little interruption to your databases, provided you plan the move well.



On April 17, 2018, we finalized the hardware and software configurations we tested. Updates for current and recently released hardware and software appear often, so unavoidably these configurations may not represent the latest versions available when this report appears. For older systems, we chose configurations representative of typical purchases of those systems. We concluded hands-on testing on May 22, 2018.

System configuration information

The table below presents detailed information on the systems we tested.

Server configuration information	Dell EMC PowerEdge R740xd
BIOS name and version	Dell 1.3.7
Non-default BIOS settings	None
Operating system name and version/build number	Windows Server® 2016 Standard, Version 1607, Build 14393.2155
Date of last OS updates/patches applied	03/19/18
Power management policy	Performance
Processor	
Number of processors	2
Vendor and model	Intel® Xeon® Gold 6130
Core count (per processor)	16
Core frequency (GHz)	2.10
Stepping	H0
Memory module(s)	
Total memory in system (GB)	192
Number of memory modules	12
Vendor and model	Hynix HMA82GR7AFR8N-VK
Size (GB)	16
Type	DDR4-2666
Speed (MHz)	2,666
Speed running in the server (MHz)	2,666
Storage controller	
Vendor and model	Dell PERC H730P Mini
Cache size (GB)	1GB
Firmware version	25.5.4.0006
Driver version	6.604.06.00
Local storage (type A)	

Server configuration information	Dell EMC PowerEdge R740xd
Number of drives	2
Drive vendor and model	Dell ST300MP0026
Drive size (GB)	300
Drive information (speed, interface, type)	15K, 12Gbps, SAS, HDD
Local storage (type B)	
Number of drives	12
Drive vendor and model	Dell THNSF8960CCSE
Drive size (GB)	960
Drive information (speed, interface, type)	6Gbps, SATA, SSD
Network adapter	
Vendor and model	QLogic BCM57800 10 GbE
Number and type of ports	2 x 1GbE, 2 x 10GbE
Driver version	7.13.104.0
Cooling fans	
Vendor and model	Nidec UltraFlo 4VXP3-X30
Number of cooling fans	6
Power supplies	
Vendor and model	Dell D1600E-S0
Number of power supplies	2
Wattage of each (W)	1,600

Installing the SQL Server environment

Configuring the Dell EMC PowerEdge R740xd internal storage

1. During POST, press Ctrl-R at the prompt to enter the PERC Configuration Utility.
2. Select the controller, and press F2. Select Create VD.
3. Specify the RAID level, select the desired physical disks, choose the cache settings required, and press Apply.
4. Exit the configuration utility, and reboot the server.

Installing Microsoft Windows Server 2016

1. Insert the Microsoft Windows Server 2016 installation DVD into the optical drive.
2. Power on the server, and boot into the Windows installer.
3. Once the installation files have been copied into memory, select the appropriate language, time and currency format, and keyboard layout, and click Next.
4. Click Install Now.
5. Enter the product key, and click Next.
6. Choose Windows Server 2016 Datacenter (Server with a GUI), and click Next.
7. Check the I accept the license terms box, and click Next.
8. Choose Custom: Install Windows only (advanced).
9. Select the OS hard drive from the list, and click Next. This will begin the installation.

Installing Microsoft SQL Server 2017

1. Using Server Manager, add the .NET Framework 4.5 feature, and reboot if necessary.
2. Attach the Microsoft SQL Server 2017 installation media to the server.
3. In the guest operating system, run SETUP.EXE.
4. Select Installation, and click New SQL Server stand-alone installation or add features to an existing installation.
5. Enter your product key, and click Next.
6. Check the I accept the license terms checkbox, optionally check the Customer Experience Improvement Program and Error Reporting checkbox, and click Next.
7. Allow the Global Rules check to complete. If there are no errors, check the Use Microsoft Update to check for updates (recommended) checkbox, and click Next.
8. After the Product Updates, Install Setup Files and Install Rules processes complete, select the SQL Server Feature Installation radio button, and click Next.
9. At the Feature Selection screen, select the features required by your organization. For this guide, we selected Database Engine Services, Full-Text and Semantic Extractions for Search, Client Tools Connectivity, Client Tool Backwards Compatibility, Management Tools – Basic, and Management Tools – Complete. Specify a directory (we kept the defaults), and click Next.
10. Choose Default instance or Named instance depending on the needs of your organization (we chose Default instance). Click Next.
11. On the Server Configuration screen, choose the accounts to be used for each SQL Server component. We left the default accounts. Click Next.
12. On the Database Engine Configuration screen, address the following:
 - a. On the Server Configuration tab, select Windows Authentication mode or Mixed Mode. Add any necessary users and provide a password for the sa account if using Mixed Mode. We selected Mixed Mode and added the current user (Windows Local Administrator).
 - b. On the Data Directories tab, enter the default directories where SQL Server installation will attempt to place its user database and log files upon creation.
 - c. TempDB: Enter the number of files, initial size, and default directories for both the TempDB data and TempDB data log files, and set an appropriate Autogrowth value.
 - d. On the FILESTREAM tab, enable FILESTREAM if appropriate for your organization. We left FILESTREAM disabled.
13. Click Next.
14. At the Ready to Install screen, review the summary info, and click Install.
15. After the installation completes, click Close.

Backing up and restoring the database

Moving the SQL Server 2008 R2 database

Performing the backup

1. Using SQL Server Management Studio, connect to your SQL Server 2008 R2 instance.
2. Right-click the database, and click Properties.
3. In the left pane, select Options.
4. Either set the database to read-only mode or set the access property to SINGLE_USER. Both choices immediately sever all user connections.
5. Click OK.
6. Right-click the database, and click Tasks→Backup.
7. Provide the necessary options for the backup. For this guide, we chose a Full backup type and backed up the database to a file on local storage. We left the default Media and Backup Options.

Restoring the database to the SQL Server 2017 instance

1. Open the SQL Server Management Studio.
2. Using SQL Server Management Studio, connect to your SQL Server 2017 instance.
3. Right-click Databases, and click Restore Database.
4. If restoring from a file, select the radio button for Device and browse to the backup file. Change the new file locations if necessary, and provide any further options. To begin the restore process, click OK.
5. Right-click the newly-restored database, and click Properties.
6. In the left pane, select Options.
7. Set the database name to be the same as the legacy database (by default it will have the same name). Change the SQL Server compatibility level to 2017 (level 140). Change the database to read-write and/or set the access level to MULTI_USER.
8. Click OK.
9. Modify application connection strings or change Domain Name System (DNS) pointers that reference the migrated database.

Moving the SQL Server 2008 R2 database using Microsoft Data Migration Assistant

Using the Migration option in Microsoft Data Migration Assistant

1. Open Data Migration Assistant.
2. Click the + button.
3. In the New window, change the Project type to Migration, your Source server type to SQL Server, and your Target server type to SQL Server, and click Create.
4. In the Specify source & target window, put your 2008 R2 SQL Server as your source and your 2017 SQL Server as your target, filling in authentication details as appropriate, and click Next.
5. In Add databases, check the appropriate databases, verify the destination locations for them, and click Next.
6. In Select logins, make sure to copy your necessary logins for your databases, and click Start Migration.

Post-installation tasks

1. Log into the host.
2. In SQL Server Management Studio, in the Object Explorer pane, connect to both your SQL Server 2008 R2 server and your SQL Server 2017 server. Be sure to have the Object Explorer Details tab open (View→Object Explorer Details).
3. Expand the tree view of the SQL Server 2008 R2 server, browse to the Security tab, and click the Logins node.
4. Select the logins you wish to migrate.
5. Right-click the logins you selected, and choose Script Login As→Create To→New Query Window.
6. Right-click the query window, and click Connection→Change Connection.
7. Choose to connect to your SQL Server 2017 instance.
8. Execute the script on your SQL Server 2017 server to create the logins. If you are transferring Windows-based logins, the process is complete. If you are transferring SQL Server logins, continue with step 7.
9. For SQL Server logins, the script you executed in step 6 creates the login, marks it as disabled, and assigns it a random password, but the script does not map the login to the database user. To avoid having a database user that is "orphaned" from a login, use the `sp_change_users_login` stored procedure to view orphaned users and to map a user to a login.
10. In the left pane, expand SQL Server Agent.
11. Locate the jobs you need to migrate in SQL Server Management Studio (on your SQL Server 2008 R2 server), and choose Script Job As→Create To→New Query Window.
12. Connect to the SQL Server 2017 server, and run the resulting script in a query window. Verify the job was created successfully in SQL Server Management Studio.

This project was commissioned by Dell Technologies.



Facts matter.®

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners.

DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY:

Principled Technologies, Inc. has made reasonable efforts to ensure the accuracy and validity of its testing, however, Principled Technologies, Inc. specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Principled Technologies, Inc., its employees and its subcontractors shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result.

In no event shall Principled Technologies, Inc. be liable for indirect, special, incidental, or consequential damages in connection with its testing, even if advised of the possibility of such damages. In no event shall Principled Technologies, Inc.'s liability, including for direct damages, exceed the amounts paid in connection with Principled Technologies, Inc.'s testing. Customer's sole and exclusive remedies are as set forth herein.