



Inside **OUT**

The ultimate in-depth reference
Hundreds of timesaving solutions
Supremely organized, packed
with expert advice

SQL Server 2019 Administration

Randolph West ▪ Melody Zacharias ▪ William Assaf
Sven Aelterman ▪ Louis Davidson ▪ Joseph D'Antoni

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



SQL Server 2019 Administration

Randolph West
Melody Zacharias
William Assaf
Sven Aelterman
Louis Davidson
Joseph D'Antoni

SQL Server 2019 Administration Inside Out
Published with the authorization of Microsoft Corporation by:
Pearson Education, Inc.

Copyright© 2020 by Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearson.com/permissions

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-13-556108-9

ISBN-10: 0-13-556108-6

Library of Congress Control Number: 2020930782

ScoutAutomatedPrintCode

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author, the publisher, and Microsoft Corporation shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the programs accompanying it.

To Marinus (obviously), but also Larry Toothman, Robert Davis, and Ian van Schalkwyk, because I miss you.

– Randolph West

To my husband Chris and daughters Jessica and Chloe for all your patience and understanding.

– Melody Zacharias

To Christine from her biggest fan

– William Assaf

In Memory of Erick Aelterman

Your example of hard work and perseverance made this possible

– Sven Aeletterman

To the rest of the writing and tech editing team. This edition of the book has been so much more of a joy due to everyone's hard work and commitment to the project. I am very proud to have been chosen to not only tech-edit half of this book, but to also take charge of revising two chapters.

– Louis Davidson

I'd like to thank the SQL Server Product Group for their hard work on the product.

– Joseph D'Antoni

To my dog Buster. Thanks for being patient and dealing with fewer dog park visits while I edited this book.

– Meagan Longoria



Contents at a Glance

Part I Introduction

Chapter 1	
Getting started with SQL Server tools	3
Chapter 2	
Introducing database server components	53
Chapter 3	
Designing and implementing an on-premises database infrastructure	91

Part II Deployment

Chapter 4	
Installing and configuring SQL Server instances and features	141
Chapter 5	
Installing and configuring SQL Server on Linux	189
Chapter 6	
Provisioning and configuring SQL Server databases	209
Chapter 7	
Understanding table features	243

Part III SQL Server management

Chapter 8	
Maintaining and monitoring SQL Server	307
Chapter 9	
Automating SQL Server administration	381
Chapter 10	
Developing, deploying, and managing data recovery	433
Chapter 11	
Implementing high availability and disaster recovery	471

Part IV Security

Chapter 12	
Administering security and permissions	539
Chapter 13	
Protecting data through encryption, privacy and auditing	605

Part V Performance

Chapter 14	
Performance tuning SQL Server	657
Chapter 15	
Understanding and designing indexes	741

Part VI Cloud

Chapter 16	
Designing and implementing hybrid and Azure database infrastructure	779
Chapter 17	
Provisioning Azure SQL Database	819
Chapter 18	
Provisioning Azure SQL Database managed instance	855
Chapter 19	
Migrating to SQL Server solutions in Azure	881

Part VII Big Data and Machine Learning

Chapter 20	
Leveraging big data and machine learning	905
Index	937



Table of Contents

	About the authors	xix
	Introduction	xxi
	Who this book is for	xxii
	How this book is organized	xxii
	Acknowledgments	xxvi
	Errata, updates, & book support	xxvii
Part I	Introduction	1
Chapter 1	Getting started with SQL Server tools	3
	SQL Server setup	3
	Installing SQL Server by using the Installation Center	4
	Planning before an upgrade or installation	5
	Installing or upgrading SQL Server	9
	Tools and services installed with the SQL Server Database Engine	9
	Machine Learning Services	10
	Data Quality Services	11
	Command line interface	12
	SQL Server Configuration Manager	15
	Performance and reliability monitoring tools	16
	Database Engine Tuning Advisor	16
	Extended events	17
	Management data warehouse	19
	SQL Server Reporting Services	22
	Installation	22
	Report Services Configuration Manager	24
	SQL Server Management Studio (SSMS)	24
	Releases and versions	25
	Installing SQL Server Management Studio	25
	Upgrading SQL Server Management Studio	26
	Features of SQL Server Management Studio	26
	Additional tools in SQL Server Management Studio	32
	Error logs	35
	Activity Monitor	36
	SQL Server Agent	40
	Azure Data Studio	44
	User interface	44

	Highlighted features in Azure Data Studio	46
	Notebooks in Azure Data Studio	48
	SQL Server Data Tools	49
	SQL Server Integration Services	49
	A note on discontinued and deprecated features	52
Chapter 2	Introducing database server components	53
	Memory	54
	Understanding the working set	54
	Caching data in the buffer pool	54
	Caching plans in the procedure cache	56
	Lock pages in memory	56
	Editions and memory limits	57
	Central Processing Unit	58
	Simultaneous multithreading	58
	Non-Uniform Memory Access	60
	Disable power saving everywhere	61
	Storing your data	61
	Types of storage	62
	Configuring the storage layer	64
	Connecting to SQL Server over the network	69
	Protocols and ports	70
	Added complexity with Virtual Local Area Networks	70
	High availability concepts	70
	Why redundancy matters	72
	Disaster recovery	72
	Clustering	73
	The versatility of log shipping	76
	Always On availability groups	76
	Read-scale availability groups	79
	Distributed availability groups	79
	Basic availability groups	79
	Securing SQL Server	80
	Integrated authentication and Active Directory	80
	Azure Active Directory	83
	Understanding virtualization and containers	85
	Going virtual	85
	Resource provisioning for virtual consumers	86
	When processors are no longer processors	88
	The network is virtual, too	89
Chapter 3	Designing and implementing an on-premises database infrastructure 91	
	Introduction to SQL Server database architecture	91
	Data files and filegroups	92
	Grouping data pages with extents	93
	Contents and types of data pages	94
	Verifying data pages by using a checksum	96

Recording changes in the transaction log	97
Flushing data to the storage subsystem with checkpoints	98
Inside the transaction log file	98
The Minimum Recovery LSN	101
Types of database checkpoints	101
Restarting with recovery	103
MinLSN and the active log	104
A faster recovery with Accelerated Database Recovery	104
Table partitioning	105
Compressing data	107
Table and index compression	107
Backup compression	109
Managing the temporary database	110
Storage options for TempDB	110
Recommended number of files	111
Configuration settings	112
Managing system usage with Resource Governor	112
Configuring the operating system page file	113
Taking advantage of logical processors by using parallelism	114
SQL Server memory settings	116
Allocating CPU cores with an affinity mask	120
File system configuration	122
Container orchestration with Kubernetes	125
Kubernetes support for SQL Server	126
Deploying SQL Server in containers	128
Getting started with SQL Server on Kubernetes	131
Deploying SQL Server on Kubernetes	132
Reviewing cluster health	136
Part II Deployment	139
Chapter 4 Installing and configuring SQL Server instances and features	141
What to do before installing SQL Server	141
Deciding on volume usage	142
Important SQL Server volume settings	145
SQL Server editions	147
Installing a new instance	149
Planning for multiple SQL Server instances	149
Creating an Azure VM running SQL Server	150
Installing SQL Server on Windows	151
Installing common features	161
Logging SQL Server Setup	166
Automating SQL Server Setup by using configuration files	167
Post-installation server configuration	171
Post-installation checklist	171
Post-installation configuration of other features	184
SSISDB initial configuration and setup	184

	SQL Server Reporting Services initial configuration and setup	185
	SQL Server Analysis Services initial configuration and setup	188
Chapter 5	Installing and configuring SQL Server on Linux	189
	What is Linux?	189
	Differences between Windows and Linux	190
	Linux distributions supported by SQL Server	193
	Considerations for installing SQL Server on Linux.	194
	Configuring OS settings.	194
	Installing SQL Server on Linux	197
	Installation requirements	198
	Downloading and installing packages	198
	Configuring SQL Server for Linux	200
	Using mssql-conf to set up and configure SQL Server	201
	Caveats of SQL Server on Linux	206
	Missing SQL Server features on Linux	207
Chapter 6	Provisioning and configuring SQL Server databases.	209
	Adding databases to a SQL Server instance	210
	Creating a database	210
	Moving existing databases	214
	Upgrading database compatibility levels.	216
	Other Considerations for migrating databases	219
	Database properties and options	223
	Moving and removing databases	234
	Moving user and system databases	234
	Database actions: offline versus detach versus drop	236
	Single-user mode.	240
Chapter 7	Understanding table features	243
	Reviewing table design	243
	General purpose data types	244
	Specialized data types	252
	Data type precedence	260
	Constraints	260
	Sequence objects.	263
	User-defined data types and user-defined types	267
	Sparse Columns	268
	Computed columns.	269
	Special table types.	270
	System-versioned temporal tables.	270
	Memory-optimized tables	276
	Graph tables	281
	Storing BLOBs	287
	Understanding FILESTREAM.	288
	FileTable	290

Table partitioning	290
Horizontally partitioned tables and indexes	291
Vertical partitioning	297
Capturing modifications to data	298
Using change tracking	299
Using change data capture	301
Querying change tracking and change data capture	302
Comparing change tracking, change data capture, and temporal tables	304
Part III SQL Server management	305
Chapter 8 Maintaining and monitoring SQL Server	307
Detecting, preventing, and responding to database corruption	307
Setting the database's page verify option	308
Using DBCC CHECKDB	308
Repairing database data file corruption	311
Recovering from database transaction log file corruption	312
Database corruption in databases in Azure SQL Database	312
Maintaining indexes and statistics	313
Changing the Fill Factor property when beneficial	314
Monitoring index fragmentation	315
Maintaining Indexes	317
Managing database file sizes	326
Understanding and finding autogrowth events	328
Shrinking database files	329
Monitoring activity by using DMOs	331
Observing sessions and requests	331
Understanding wait types and wait statistics	334
Monitoring with the SQL Assessment API	343
Utilizing extended events	347
Viewing extended events data	348
Using extended events to capture deadlocks	353
Using extended events to detect autogrowth events	355
Using extended events to detect page splits	356
Securing extended events	357
Capturing performance metrics with DMOs and Data Collectors	357
Querying performance metrics by using DMVs	358
Capturing performance metrics by using Performance Monitor	361
Monitoring key performance metrics	362
Monitoring key performance metrics in Linux	366
Monitoring key performance metrics in Azure Data Portal	367
Protecting important workloads by using Resource Governor	372
Configuring the Resource Governor classifier function	373
Configuring Resource Governor pools and groups	374
Monitoring Resource Pools and Workload Groups	376
Understanding the new servicing model	377

Chapter 9	Automating SQL Server administration	381
	Components of SQL Server automated administration	382
	Database Mail	382
	SQL Server Agent	387
	Maintaining SQL Server	400
	Basic “care and feeding” of SQL Server	400
	Using SQL Server Maintenance Plans	402
	Covering databases with the Maintenance Plan	403
	Maintenance plan tasks	404
	Maintenance Plan report options	412
	Building Maintenance Plans using the designer in SSMS	412
	Backing up availability groups using a secondary replica	414
	Strategies for administering multiple SQL Servers	416
	Master and Target servers for SQL Agent jobs	416
	SQL Server Agent event forwarding	419
	Policy-Based Management	420
	Using PowerShell to automate SQL Server administration	422
	PowerShell basics	424
	Installing the PowerShell SQLServer module	425
	Using PowerShell with SQL Server	427
	Using PowerShell with availability groups	431
Chapter 10	Developing, deploying, and managing data recovery	433
	Preparing for data recovery	434
	A disaster recovery scenario	435
	Defining acceptable data loss: RPO	437
	Defining acceptable downtime: RTO	438
	Establishing and using a run book	438
	Understanding different types of backups	439
	An overview of SQL Server recovery models	440
	Backup types	442
	The backup chain	445
	File and filegroup backups	447
	Additional backup options and considerations	448
	Availability and security of backup media	451
	Understanding backup devices	452
	Backup to disk	452
	Backup to URL	452
	Backup and media sets	454
	Creating and verifying backups	456
	Creating backups	456
	Verifying backups	458
	Restoring a database	459
	Restoring a database using a full backup	460
	Restoring a database with differential and log backups	460
	Restoring a database to a point in time	462

	Restoring a database piecemeal	464
	Defining a recovery strategy	466
	A sample recovery strategy for our DR scenario	466
	Recovery strategies for hybrid and cloud environments	469
Chapter 11	Implementing high availability and disaster recovery	471
	Overview of high availability and disaster recovery technologies in SQL Server	471
	Comparing HA and DR technologies	472
	Understanding log shipping	474
	Understanding types of replication	476
	Understanding the capabilities of failover clustering	480
	Understanding the capabilities of availability groups	483
	Configuring failover cluster instances	485
	Understanding FCI quorum	485
	Configuring a SQL Server FCI	488
	Patching A Failover Cluster	490
	Designing availability groups solutions	491
	Comparing different cluster types	492
	Creating WSFC for use with availability groups	496
	Understanding the database mirroring endpoint	497
	Improvements to availability groups recently	498
	Choosing the correct secondary replica availability mode	501
	Understanding the impact of secondary replicas on performance	502
	Understanding failovers in availability groups	504
	Seeding options when adding replicas	508
	Additional actions after creating an availability group	512
	Reading secondary database copies	514
	Implementing a hybrid availability group topology	521
	Configuring availability groups in SQL Server on Linux	522
	Understanding the differences between Windows and Linux clustering	522
	Setting up an availability group in SQL Server on Linux	523
	Administering availability groups	529
	Analyzing DMVs for availability groups	529
	Analyzing wait types for availability groups	532
	Analyzing extended events for availability groups	533
	Alerting for availability groups	535
Part IV	Security	537
Chapter 12	Administering security and permissions	539
	Authentication modes	539
	Windows authentication	540
	SQL Server Authentication	541
	Azure Active Directory universal authentication	541
	Azure Active Directory password authentication	542
	Azure Active Directory integrated authentication	542

Advanced types of server principals	542
Authentication to SQL Server on Linux	543
Contained database authentication	544
Security principals	544
Understanding the basics of privileges	546
Configuring login server principals	548
Configuring database principals	567
Permissions	578
Understanding permissions for Data Definition Language and Data Manipulation Language	578
Understanding overlapping permissions	580
Understanding authorization	582
Common security administration tasks	591
Solving orphaned SIDs	591
Preventing orphaned SIDs	594
Moving SQL Server logins and permissions	594
Saving the day with the Dedicated Administrator Connection	601

Chapter 13 Protecting data through encryption, privacy and auditing605

Privacy in the modern era	605
General Data Protection Regulation (GDPR)	606
Introducing security principles and protocols	607
Securing your environment with defense in depth	607
The difference between hashing and encryption	609
A primer on protocols and transmitting data	612
Symmetric and asymmetric encryption	617
Digital certificates	618
Protecting the data platform	619
Securing the network with TLS	620
Data protection from the OS	622
The encryption hierarchy in detail	622
Using EKM modules with SQL Server	623
Master keys in the encryption hierarchy	625
Encrypting data by using TDE	627
Protecting sensitive columns with Always Encrypted	631
Row-level security	637
Dynamic data masking	638
Protecting Azure SQL Database	639
Auditing with SQL Server and Azure SQL Database	641
SQL Server Audit	641
Auditing with Azure SQL Database	648
Securing Azure infrastructure as a service	648
Network Security Group	649
User-defined routes and IP forwarding	650
Additional security features in Azure networking	652

Part V	Performance	655
Chapter 14	Performance tuning SQL Server	657
	Understanding isolation levels and concurrency	658
	Understanding how concurrent sessions become blocked	663
	Changing the isolation level	667
	Understanding and handling common concurrency scenarios	669
	Understanding the enterprise solution to concurrency: row version-based concurrency	680
	Understanding on-disk versus memory-optimized concurrency	688
	Understanding durability settings for performance	691
	Delayed durability database options	692
	Understanding how SQL Server executes your query	694
	Understanding the overall query execution process	694
	Retrieving execution plans in SQL Server Management Studio	696
	Understanding execution plans	702
	Understanding parameterization and “parameter sniffing”	713
	Understanding the Procedure Cache	715
	Understanding parallelism	720
	Understanding advanced engine features for tuning queries	722
	Plan Guides and Query Store	723
	Automatic Plan Correction	732
	Intelligent Query Processing	733
Chapter 15	Understanding and designing indexes	741
	Designing clustered indexes	742
	Choosing a proper rowstore clustered index key	742
	The case against intentionally designing heaps	746
	Understanding the OPTIMIZE_FOR_SEQUENTIAL_KEY feature	747
	Designing rowstore nonclustered indexes	748
	Understanding nonclustered index design	749
	Creating filtered nonclustered indexes	755
	Understanding the “missing” indexes feature	756
	Understanding and proving index usage	760
	Understanding columnstore indexes	761
	Designing columnstore indexes	763
	Understanding batch mode	764
	Understanding the deltastore of columnstore indexes	765
	Demonstrating the power of columnstore indexes	766
	Understanding indexing in memory-optimized tables	769
	Understanding hash indexes for memory-optimized tables	770
	Understanding nonclustered indexes for memory-optimized tables	771
	Understanding index statistics	771
	Automatically creating and updating statistics	772
	Manually creating statistics for on-disk tables	773

	Understanding statistics on memory-optimized tables	773
	Understanding statistics on external tables	774
	Understanding other types of indexes	774
	Understanding full-text indexes	774
	Understanding spatial indexes	775
	Understanding XML indexes	775
Part VI	Cloud	777
Chapter 16	Designing and implementing hybrid and Azure database infrastructure	779
	Cloud computing and Microsoft Azure	779
	Database-as-a-service	780
	Managing Azure: The Azure portal and PowerShell Core	780
	Azure governance	782
	Cloud-first	784
	Resource scalability	784
	Networking in Azure	785
	Cloud models and SQL Server	786
	Infrastructure-as-a-service	787
	Platform-as-a-service	795
	Hybrid cloud with Azure	809
	Cloud security	816
	Other data services in Azure	816
Chapter 17	Provisioning Azure SQL Database	819
	Provisioning a logical SQL server	820
	Creating a server using the Azure portal	821
	Creating a server by using PowerShell Core	822
	Establishing a connection to your server	823
	Deleting a server	825
	Provisioning a database in Azure SQL Database	825
	Creating a database using the Azure portal	826
	Creating a database by using PowerShell Core	827
	Creating a database by using Azure CLI	828
	Creating a database by using T-SQL	829
	Scaling up or down	829
	Provisioning an elastic pool	830
	Managing database space	831
	Security in Azure SQL Database	832
	Security features shared with SQL Server 2019	832
	Server and database-level firewall	832
	Integration with virtual networks	835
	Access control using Azure AD	835
	Role-Based Access Control	837
	Auditing	838

Advanced Data Security	842
Preparing Azure SQL Database for disaster recovery	845
Understanding default disaster recovery features	845
Manually backing up a database	846
Enabling zone redundant configuration	849
Configuring geo-replication	849
Setting up failover groups	852
Using Azure Backup for long-term backup retention	854
Chapter 18 Provisioning Azure SQL Database managed instance	855
What is a Managed Instance?	855
Differences between SQL Server on-premises and in a Managed Instance	856
Creating a managed instance	860
Selecting a pricing tier and service objective	861
Provisioning a managed instance from the portal	862
Provisioning a managed instance using PowerShell	863
Establishing a connection	865
Creating the endpoints via the portal	866
Creating the VPN Gateway via PowerShell	866
Connect from an on-premises computer	867
Connect from a Virtual Machine (VM)	868
Migrating data to Managed Instance	869
Data Migration Service	869
Backup and restore	870
Deleting a Managed Instance	871
Network requirements for Managed Instances	872
Subnet	872
Network Security Group	872
User Defined Route table	873
Endpoints	873
IP addresses	873
Managed Instance administration features	873
High Availability	874
Replication	875
Scaling up or down	876
Automated backups	876
Azure SQL Analytics	877
Security of Managed Instance	877
Azure Active Directory (Azure AD)	877
Data protection features	879
Isolation	879
Auditing	879
Data encryption	879
Row-level security	880
Chapter 19 Migrating to SQL Server solutions in Azure	881
Migration Services Options	881

	Database Experimentation Assistant (DEA)	882
	Azure Data Migration Assistant (DMA)	884
	Specifics for Linux migrations	885
	Features of Data Migration Assistant	885
	Database Migration Service	886
	SQL Server Migration Assistant	888
	Common causes for migration failures	889
	Open Source PowerShell migration option	891
	Integration Runtime options for Migration Services	896
	Best Practices for security and resiliency during migration	899
Part VII	Big Data and Machine Learning	903
Chapter 20	Leveraging big data and machine learning	905
	A big data primer	906
	Introducing MapReduce with Hadoop and HDFS	906
	Introducing distributed memory with Spark	907
	Introducing Big Data Clusters	907
	Big Data Clusters technical overview	909
	Deploying Big Data Clusters	912
	Configuring BDC storage	913
	Big Data Clusters security	913
	Benefits of PolyBase	914
	Unified data platform features	915
	Operationalizing analytics with Machine Learning Server	925
	Architecture	926
	Machine Learning Server	928
	How to operationalize your models	929
	Launchpad service	933
	Index	937

About the authors



Randolph West (they / them) is a technologist and Data Platform MVP from Calgary, Alberta, Canada, and hates writing in the third person because pronouns suck. Randolph is the founder of the Calgary Data User Group, and leader of the Calgary PASS User Group. You can see Randolph presenting at various conferences around the world, and acting on stage and screen. Randolph specializes in implementing best practices, performance tuning, disaster recovery, cloud migrations, software development, and yelling at Microsoft Word. If you're interested in becoming a public speaker, check out SpeakingMentors.com. Blog: bornsql.ca/blog. Twitter: [@_randolph_west](https://twitter.com/_randolph_west). Not to be trusted around chocolate.



Melody Zacharias has worked with SQL Server since version 7, and she is very passionate about all things data! As a Microsoft MVP Melody loves to share her passion with users around the world by teaching at events, working within the SQL community, and mentoring. When not working with data she loves to hike with her dog in the Canadian Rockies. You can find her on her blog at SQLMelody.com where she shares musings on how to make SQL sing, or sometimes just carry a tune, or on Twitter [@SQLMelody](https://twitter.com/SQLMelody).



William Assaf (he/him) is a Microsoft SQL Server consultant, speaker, and manager, and blogs about SQL at sqltact.com. William has been a developer, admin, and consultant for databases around the world, having launched from in-house DBA to consulting DBA thanks to the recession. He has helped develop Microsoft SQL Server certification exams since 2012, has been a Regional Mentor for PASS since 2015, and just stepped down from being the lead organizer of SQLSaturday Baton Rouge after a decade leading one of the largest SQLSat events in the world, and the only one with jambalaya. William enjoys volunteering for STEM initiatives as well as guest lecturing at LSU and SELU, where he gives a shoutout to all the ISDS and CIS students. William lives in continuous awe of his far more accomplished high school sweetheart Christine, and they would like to say that empty nesting is pretty great. Both of them hope to see you at PASS Summit and a future SQLSaturday event.



Sven Aelterman started with SQL Server when he first deployed version 2000 in a failover cluster scenario. Since then, he has worked as IT manager, principal consultant, and IT director. He currently serves the Trojans (students) of Troy University as a lecturer in information systems in the Sorrell College of Business and as director of IT for the College. In addition, he is cloud software architect for Sorrell Solutions, a business services nonprofit through which Trojans can gain real-world business and IT experience. In a fledgling attempt to give back to the community, he has spoken at many SQL Saturdays and code camps in the southeastern United States since 2005. He spoke about SSIS 2012 at Microsoft TechEd 2012. In 2012, he co-authored a book dedicated to SQL Server FILESTREAM. He was a co-author on the 2017 edition of this text. His involvement with Microsoft Azure resulted in the organization of four Global Azure Bootcamp events at Troy University. Sven blogs about a variety of Microsoft technologies at blog.aelterman.com and tweets and retweets about technology [@svenaelterman](https://twitter.com/svenaelterman).



Louis Davidson is a data architect for CBN in Virginia Beach, VA; telecommuting from Cleveland, TN (which is not even as glamorous as it sounds.) Louis has written and contributed to many books on SQL Server topics over the past 20 years. His most prominent work has been five editions of his book entitled: “Pro SQL Server Relational Database Design and Implementation” for Apress in 2016, with a new version forthcoming in 2020. Louis has been a speaker at many SQL Saturday events, and has helped organize events in Nashville and Chattanooga, TN. He blogs on Red-Gate’s Simple-Talk website here: <https://www.red-gate.com/simple-talk/author/louis-davidson/> and you can visit his website at <http://drsqli.org>.



Joseph D’Antoni is a Principal Consultant and Microsoft Data Platform MVP with over 20 years of experience working in both Fortune 500 and smaller firms. He has worked with SQL Server and Oracle since version 7 of each database and has specific expertise in performance tuning, infrastructure, and disaster recovery.



Meagan Longoria is a Microsoft Data Platform MVP living in Denver, Colorado. She is an experienced consultant who has worked in business intelligence, data warehousing, and database development for over a decade. She enjoys creating solutions in Azure and SQL Server that make data useful for decision makers. Meagan enjoys sharing her knowledge with the technical community by speaking at conferences, blogging (Datasavvy.me), and sharing tips and helpful links on twitter (@mmarie).

Introduction

Much has changed in the months leading up to the publishing of Microsoft SQL Server 2019 Administration Inside Out, a follow-up edition to this team's successful and well-reviewed Microsoft SQL Server 2017 Administration Inside Out. Among just our writing and editing team, one became an empty nester, another got a new knee, another two new hips. One was evacuated because of forest fires, another left powerless under a hurricane, another twice moved houses, others got new jobs and job roles. Countless naming conventions and product names were changed in the Microsoft data platform, too, causing many late edits to this book. Azure SQL Database managed instance was made generally available, as well as the concepts of Big Data Clusters, Azure DevOps, Azure SQL Database Hyperscale and Azure SQL Database serverless tier, Azure Arc, Azure Cosmos DB, and more. SQL Server 2008 and 2008 R2 hit their end of life support dates and are no longer supported by Microsoft*, new streaming services were launched, the Washington Nationals won the World Series, LSU beat Alabama in football, and Avengers: Endgame finally beat Avatar as the highest grossing film of all time... so yeah, a lot of long-standing states of information have changed.

While no book can keep up with all the changes Microsoft has made even to just the data platform and supporting technologies in Azure, this book does its best to fully outer join the latest guidance on modern data platform solutions with long-standing database administrator wisdom. We'll cover both columnstore and rowstore indexes. Both Linux and Windows. Both Azure and on-prem solutions (and hybrid). Both memory-optimized tables and disk-based tables, with optimistic and pessimistic isolation. This book aims to be a reference, inspiration, early radar warning, or at least a launch pad, for your continued education and data platform solutions development.

Data warehousing has changed, with the traditional extract-transform-load (ETL) strategies giving way to extract-load-transform on-demand (ELT) strategies. SQL Server 2019 has new features and tools to develop these solutions while leveraging your existing Transact-SQL skillset. Features like Big Data Clusters and PolyBase external tables are helping you design fast, scalable, modern solutions where once you had to rely on external products, custom solutions in SQL Server Integration Services (SSIS), or linked server objects. All these solutions have their place, and all will continue to be in the inventory of your DBA utility belt, but your ensured success as a modern, relevant DBA requires adding tools. This book, and its large accompanying repository of sample scripts, aims to be that reference to DBAs as they add modern tools to their skill set.

Threat vectors and regulatory requirements have also changed. That's why SQL Server 2019 has vastly improved data classification capabilities leveraging a new dedicated schema, as well as Always Encrypted now with secure enclaves. There's also a new certificate manager built-in to SQL Server Configuration Manager, and new integrations with Azure Key Vault. We've expanded

the SQL Server security and protecting data chapters greatly from the last book, recognizing the DBA's role as key to an enterprise data security team.

This book's content, and especially the Inside OUT features, are derived directly from the skills and experience of a group of seasoned database professionals with many decades of experience in designing, optimizing, developing, and teaching complex modern data platform solutions. It is written for experienced DBAs and developers.

After reading this book, you will be able implement a cloud-based, hybrid, and on-premises data solution using SQL Server 2019.

The velocity of change for the Microsoft SQL Server DBA has increased this decade. Gone are the days when DBAs had between three and five years to soak in and adjust to new tools and tricks in the Database Engine and surrounding features. We had less than a calendar year between the release of SQL Server 2017 and the first technical previews of SQL Server 2019.

This book is written and edited by SQL Server experts with two goals in mind: to deliver a solid foundational skillset for all of the topics covered in SQL Server configuration and administration, and also to deliver awareness and functional, practical knowledge for the dramatic number of new features introduced in SQL Server 2019. We haven't avoided late-breaking content—even content that stretched the boundaries of writing deadlines with late-breaking new releases. You will be presented with not only the "how" of new features, but also the "why" and the "when" for their use.

**There are still a couple ways to get support for SQL Server 2008 / 2008 R2. One involves paying a lot of money, the other involves migrating it all to Azure VMs. Neither are as good an option as just upgrading it already!*

Who this book is for

SQL Server administration was never the narrow niche skillset that our employers might have suspected it was. Even now it continues to broaden, with support for new operating systems and platforms: cloud-based and serverless in addition to on-premises, or maybe, a little of all three. This book is for the DBAs who are unafraid to add these new skillsets and features to their utility belt, and to give courage and confidence to those who are still hesitant. SQL Server administrators should read this book to become more prepared and aware of features when talking to their colleagues in application development, data analytics, and system administration.

How this book is organized

This book gives you a comprehensive look at the various features you will use. It is structured in a logical approach to all aspects of SQL Server Administration, whether you are implementing, developing, or supporting development.

Part I: Introduction

Chapter 1, “Getting started with SQL Server tools” gives you a tour of modern tooling for SQL Server administrators, from the installation media to the free downloads, not the least of which is the rapidly evolving SQL Server Management Studio and Azure Data Studio, as well as the newly enhanced Configuration Manager, performance and reliability monitoring tools, tools for writing PowerShell, and more.

Chapter 2, “Introducing database server components” introduces the working vocabulary and concepts of database administration, starting with hardware-level topics such as memory, processors, storage, and networking. We then move into high availability basics (much more on those later), security, and hardware and OS virtualization.

Chapter 3, “Designing and implementing an on-premises database infrastructure” introduces the architecture and configuration of SQL Server, including deep dives into transaction log virtual log files (VLFs), data files, in-memory Online Transaction Processing (OLTP), and the new Accelerated Database Recovery feature, one of many new features of SQL Server 2019. We spend time with TempDB and its optimal configuration, and server-level configuration options. Finally, we introduce you to Kubernetes.

Part II: Deployment

Chapter 4, “Installing and configuring SQL Server instances and features” reviews installation of SQL Server for Windows platforms when SQL Server Setup is needed to install SQL Server. We discuss volume settings and layout for a SQL Server instance, editions, Smart Setup and unattended setup configuration, and setup logging. Look here also for post-installation checklists and configuration guidance, as well as configuration and guidance for other features including SSIS, SSAS, and SSRS, as well as PolyBase.

Chapter 5, “Installing and configuring SQL Server on Linux” reviews configuration of SQL Server on Linux instances, including feature differences between Windows and Linux. We’ll provide guidance and caveats on Linux distributions, Linux-specific monitoring and storage considerations, and tooling for setup and administration.

Chapter 6, “Provisioning and configuring SQL Server databases” reviews creation and configuration of SQL Server databases on any SQL Server platform, including strategies for migrating and moving databases. Database options and properties are discussed, as well as database collations.

Chapter 7, “Understanding table features” completes the drill down from instances to databases to tables, covering table design, data types, keys and constraints. Use of IDENTITY and sequences, computed columns and other column properties, as well as special table types are discussed. We’ll review special types of tables including temporal tables, introduce memory-optimized tables (more on these in Chapter 14), and graph tables. Finally, we’re reviewing FILESTREAM and FileTable for storing blobs, table partitioning for storing and switching large amounts of data, and strategies for tracking data changes.

Part III: SQL Server management

Chapter 8, “Maintaining and monitoring SQL Server” covers the care and feeding of SQL Server instances on both Windows and Linux, including monitoring for database corruption, monitoring index activity and fragmentation, and maintaining and monitoring indexes and index statistics. We dive into extended events, the superior alternative to traces, and also cover Resource Governor, used for insulating your critical workloads. We review monitoring and data collection strategies based in Windows, Linux, and Azure, as well as the new SQL Assessment API. We’ll finally discuss the current Microsoft servicing model for SQL Server.

Chapter 9, “Automating SQL Server administration” includes an introduction to automating activities for SQL Server, including maintenance plans, but also custom solutions involving PowerShell, including the latest features available in PowerShell. We also review built-in tools and features needed to automate tasks to your SQL Server, including database mail, SQL Server Agent jobs, Master/Target Agent jobs, proxies, SQL Agent alerts, event forwarding and Policy-Based Management.

Chapter 10, “Developing, deploying, and managing data recovery” covers the fundamentals of SQL Server database backups in preparation for disaster recovery scenarios, including a backup and recovery strategy appropriate for your environment. Backups and restores in a hybrid environment, Azure SQL Database recovery, and geo-replication are important assets for the modern DBA.

Chapter 11, “Implementing high availability and disaster recovery” then goes beyond backups and into strategies for disaster recovery from the old (log shipping and replication) to the new (availability groups), as well as monitoring and troubleshooting availability groups. We’ll compare HA and DR strategies and dive into proper architecture for maximizing SQL Server uptime.

Part IV: Security

Chapter 12, “Administering security and permissions” begins with the basics of authentication, the configuration, management, and troubleshooting of logins and users. Then, we dive into permissions, including how to grant and revoke server and database-level permissions and role membership, with a focus on moving security from server to server.

Chapter 13, “Protecting data through encryption, privacy and auditing” takes the security responsibilities of the SQL Server DBA past the basics of authentication and permissions and discusses advanced topics including the various features and techniques for encryption including Transparent Data Encryption (TDE) and Always Encrypted, as well as protecting data in motion with TLS. The new Certificate Management feature of SQL Server 2019 Configuration Manager is reviewed, as well as modern strategies for row-level security and protection of sensitive data. We discuss security measures to be taken for SQL Server instances and Azure SQL databases as well as the SQL Server Audit feature.

Part V: Performance

Chapter 14: “Performance tuning SQL Server” dives deep into isolation and concurrency options, including Read Committed Snapshot Isolation (RCSI), and why your developers

shouldn't be using NOLOCK. We'll discuss various strategies for memory-optimized data including delayed durability. We review graphical execution plans analysis, the important Query Store feature, and automatic plan correction. We review important performance-related Dynamic Management Objects (DMOs) and several new SQL Server 2019 performance features, most notably in the Intelligent Query Processing family.

Chapter 15: "Understanding and designing indexes" tackles performance from the angle of indexes, from their creation, monitoring, and tuning. We review all the various forms of indexes at our disposal, past rowstore clustered and nonclustered indexes and into other types of indexes including columnstore and memory-optimized hash. We review statistics and statistics options, including understanding how they work on a variety of index and table types.

Part VI: Cloud

Chapter 16, "Designing and implementing hybrid and Azure database infrastructure" discusses the infrastructure options for Azure-based SQL Server databases, including Platform-as-a-Service (PaaS) options of Azure SQL Database, Azure SQL managed instance, and also Infrastructure-as-a-Service options of Azure VMs running SQL Server instances. We discuss the resource scalability options for Azure SQL Database, which have dramatically expanded recently. We discuss management and governance in the Azure SQL data platform using the Azure Portal or PowerShell.

Chapter 17, "Provisioning Azure SQL Database" covers the cloud-first database service with a very high degree of compatibility with SQL Server 2019. This platform powers many web-based applications and services, scalable from basic \$5/month to 80-vCore and 100 TB. You will learn about the Azure SQL database platform, compatibility, security, and availability. You will learn how to create servers, databases, and elastic pools, and how to perform important management tasks for your databases.

Chapter 18, "Provisioning Azure SQL Database managed instance" details the powerful Azure SQL Database managed instance offering, including provisioning, managing, and scaling the instance. We review the service objectives, limitations and advantages, and security features of the managed instance.

Chapter 19, "Migrating to SQL Server solutions in Azure" covers various strategies for Azure migrations, including the Microsoft tools provided for testing and migrating SQL Server workloads. We review differences and limitations for on-prem feature migration strategies to Azure platforms, including how to migrate SSIS packages to the Integration Runtime. We review post-migration steps, best practices for security and resiliency during migration, and the common causes for migration failures.

Part VII: Big Data and Machine Learning

Chapter 20, "Leveraging Big Data and Machine Learning" starts with a primer on big data features in SQL Server, including an introduction to Big Data Clusters. We're review how to

operationalize analytics with machine learning services and the benefits of PolyBase external tables for direct access to external data sources, even non-relational data sources.

Acknowledgments

Randolph West They say it's easier the second time around. They are liars. Fortunately, I managed to rope in Melody Zacharias as payback for getting me involved in the last book. Cue evil laugh. After sitting at the kitchen counter for most of the last book, my husband bought me a desk which I never sit at. Thanks anyway! And thank you to my dog Trixie for taking me on long walks to think about better phrasing to keep Meagan happy. I would also like to give special thanks to Anthony Nocentino, Argenis Fernandez, and Glenn Berry for their help, technical advice, and support. Since we're naming names, here's a shout-out to the book team: Meagan, Melody (again), Loretta, Liv, Louis, William, Sven, and Joey. Each of you deserves a medal. This book would not be possible without the contributions of everyone involved.

Melody Zacharias I really want to thank the #SQLFamily for their inspiration, support, friendship and guidance. Particularly Argenis Fernandez who was my PASS Summit big buddy and introduced me to the community in 2011. Because of that experience, my life was forever changed for the better. No family is perfect, but my #SQLFamily is an amazingly supportive and inclusive family, and I am so proud to be a member.

William Assaf I'd like to thank the mentors and managers and colleagues in my professional career heretofore, who affected my trajectory, and to whom I remain grateful for technical and nontechnical lessons learned. I'd like to thank Connie Murla, David Alexander, Darren Schumaker, Ashagre Bishaw, Charles Sanders, Todd Howard, Chris Kimmel, Richard Caronna, Mike Huguet, Mike Carter, Jason Prell, James Sampson, my personal dream team of coauthors on this very book, and finally Patrick Leblanc, a fellow Baton Rouge native, whose friendship has repeatedly challenged and furthered me and my career. I'd also like to thank my father, a rare mechanical and electrical engineer (and a HAM), and both my older brothers who are brilliant software engineers, for repeatedly letting me play games on their computers. I'd finally like to thank Kim Fossey, Casey Phillips, Kristen Reeves, Helena Williams, Cheryl Cummings, Ajayi Anwansedo, Quinton Jason Jr, and other heroes of the STEM education in Baton Rouge, La., and all over. They are doing the hard work of developing our future coworkers and coauthors in my home state's perpetually underfunded, underappreciated, and underestimated public school youth.

Sven Aelterman There are a lot of people who have made my contributions to this book possible. They include those who have given me chances early in my career, enabled me to study abroad, and encouraged career development and growth. In no particular order, I want to recognize Walter Weyne, Yves Sucaet, Judson Edwards, Bill Belcher, Hank Findley, David Phelps, Janet and Jeff Kervin, and the late Don Hines. This "old" and expanded group of co-authors have held me to high standards, as have the editors. Speaking of editors, they're an odd breed. During the writing process: entirely unnecessary; my initial draft is perfect, of course. After the

first review round, they're know-it-all, nit-picky experts who just don't see it the way it is. By the second review round, they're usually correct. And when the book is finished, it turns out they were invaluable. Thanks, Meagan and Louis! Most importantly, my family continues to be the driving force behind what I do every day. It's not easy for Ebony, Edward, and Sofia to put up with the late nights of learning and writing. But they do it, with love and a smile (most of the time). I want them to know their support and love are valuable, valued, and acknowledged

Louis Davidson Acknowledgements are super hard, not because I am so great and have no need for others, but rather because I have been surrounded by scores of people who are amazing for so long. The authors of this and the previous edition have taught me a ton along the way. I could not have survived the writing and tech editing processes without the writers on the docs.microsoft.com site and the many blogs in the #sqlfamily community. Additionally, I must acknowledge the PASS and the Microsoft MVP program, because they have helped me to expand my horizons throughout the past 20 years, helping me to get to the point I have reached in my career.

Meagan Longoria I'd like to thank my coworkers at Denny Cherry & Associates Consulting. I learn about the Microsoft Data Platform from them every day, and I enjoy them as humans. Thanks to Melissa Coates for giving me my first technical editing opportunity, as well as for all the friendship and data platform knowledge she has shared over the years. Thanks to Melody Zacharias and Rie Irish, who gave me the opportunity to become a published author and set me on the path to more editing opportunities. Thank you to Bill Fellows, for helping me get involved in the SQL Server community, which changed my life in so many ways, and for being my friend. And of course, I'd like to thank the authors of this book: William Assaf, Randolph West, Sven Aelterman, Melody Zacharias, Louis Davidson, and Joey D'Antoni. I'm so glad to know them and thankful for everything they do in the SQL Server community.

Errata, updates, & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

[MicrosoftPressStore.com/SQLServer2019InsideOut/errata](https://microsoftpressstore.com/SQLServer2019InsideOut/errata)

If you discover an error that is not already listed, please submit it to us at the same page.

For additional book support and information, please visit:

[MicrosoftPressStore.com/Support](https://microsoftpressstore.com/Support)

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to <https://support.microsoft.com>.



PART I

Introduction

CHAPTER 1

Getting started with SQL Server tools. 3

CHAPTER 2

Introducing database server components. 53

CHAPTER 3

Designing and implementing an on-premises
database infrastructure. 91



Installing and configuring SQL Server instances and features

What to do before installing SQL Server.....	141	Post-installation server configuration	171
Installing a new instance	149	Post-installation configuration of other features	184

In this chapter, we review the process of installing and configuring a Microsoft SQL Server instance as well as the creation or migration of databases. We pay special attention to new features introduced in SQL Server 2019, and other recent features you may not have noticed in SQL Server 2017. We review a post-installation checklist for you to verify every time and, when necessary, direct you to where you can find other sources of information and details for critical steps elsewhere in this book.

The SQL Server Setup tool content of this chapter mainly applies to SQL Server installations on Windows operating systems. The installation or creation is vastly simplified for an Azure SQL Database, Azure SQL Database managed instance, SQL Server on Linux, SQL Server in Docker containers, or Azure VMs running SQL Server. Many recommended settings in this chapter still apply for server-based platforms of SQL Server, such as in Docker containers or SQL Server on Linux. They are, after all, still very much the same SQL Server product that has always existed on Windows.

In Chapter 6, “Provisioning and configuring SQL Server databases,” we’ll cover the initial creation and configuration of databases inside the SQL Server instance. This chapter, however, focuses on server-level setup and settings.

What to do before installing SQL Server

Before running the SQL Server installer on your Windows Server, there are a number of factors and settings to consider, some of which you *cannot* easily change after installation. For example, choosing between the default instance and a named instance, or choosing an instance collation, is not a choice you can easily reverse after installation. (More about the server level collation option later in this chapter, under the “Instance Collation” section.)

However, many mistakes made in installation can be resolved afterwards, likely with some tedium and outages. For example, skipping the initial default data and log directories may land all of your databases on the Operating System volume. They can be moved to the appropriate volumes later, but it’s best to get it right the first time.

CAUTION

Do not install SQL Server on the same server as a domain controller. In some scenarios, it is not supported and can even cause Setup to fail. For more information on the risks and limitations, visit: https://docs.microsoft.com/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server-ver15#DC_support.

We recommend you acquire the contents of this list prior to beginning SQL Server Setup:

- Active Directory service accounts, for the SQL Server service, SQL Agent Service, and other features if needed
- The latest downloaded cumulative update to bring the instance up to the latest patch level
- A licensing decision around the number of processors and the edition to buy
- A secure enterprise digital location for various passwords you will be generating, backups of certificates, keys
- Whether to install the default or a named instance
- A plan for where SQL Server files will go, and each volume formatted to 64KB disk unit allocation size

Let's talk more about that last item now.

Deciding on volume usage

For many good reasons, various types of SQL Server files should be placed on separate volumes. Although you can move user and system database data and log files to other locations after installation, it's best to plan your volumes prior to installation.

The examples in this chapter assume that your Windows operating system installation is on the C volume of your server. You should have many other volumes for SQL Server files—we'll review a sample layout soon. One of the basic guiding principles for a SQL Server installation is that anywhere you see "C:\," change it to another volume. This helps minimize SQL Server's footprint on the operating system (OS) volume, especially if you install multiple SQL Server instances, and can have potential disaster recovery implications in terms of volume-level backup and restores.

Inside OUT

What if you are tight on space on the OS volume after installing SQL Server?

There are some easy ways and some tricky ways to minimize the footprint of a SQL Server installation on the OS volume of your server (typically the C volume, as it is for this example). In general, SQL Server Setup and cumulative updates will delete temporary files involved in their installation, but not log files or configuration files, which should have a minimal footprint. Outside of log files, we recommend that you do not delete any files installed by SQL Server Setup or cumulative updates. Instead, let's take a look at some proactive steps to move these files off of the C volume.

Some parts of SQL Server Setup will install on the OS volume (typically, and in this and future examples, the Windows C volume). These files, which are staging areas for SQL Server Setup, are created on the OS volume in a `C:\Program Files\Microsoft SQL Server\150\Setup Bootstrap\` subfolder structure, where 150 is specific to the internal version number (15.0) of SQL 2019. This folder is used for future cumulative updates or feature changes.

If you're extremely tight on space before installing SQL Server, you will also find that the root binaries installation directory will be, by default, `C:\Program Files\Microsoft SQL Server\`. When you're using the SQL Server Setup user interface, there is no option to change this. You will, however, find this installation directory folder path listed as the `INSTANCEDIR` parameter in the config file that is generated by SQL Server Setup. How to use the config file to install SQL Server is further covered in the "Automating SQL Server Setup by using configuration files" section later in this chapter.

If this is the first SQL Server instance you are installing on a server, you will have the opportunity to change the location of shared features files, the data root directory for the instance (which contains the system databases), default database locations for user database files, and their backups. If this is not the first SQL Server 2019 instance installation on this server, the shared features directory locations (for Program Files and Program Files x86) will already be set for you, and you cannot change it.

You should place as much of the installation as possible on other volumes, not the OS volume. Keep in mind that a full-featured installation of SQL Server 2019 can consume more than 14 GB.

Inside OUT

What can you do with the D: volume on an Azure VM?

For Microsoft Azure Windows virtual machines (VMs), do not set the installation directories for any settings on the D:\ “Temporary Storage” volume. In a Linux VM, the same applies to `/dev/sdb1`.

The temporary storage volume in Azure VMs is a temporary disk that is locally present on the machine hosting your Azure VM, so it has better performance and lower latency than the default C: volume. The temporary storage volume contains only the Windows Page file by default and is wiped upon server restart, resize, or host migration!

The only possible long-term use for the temporary storage volume is for TempDB files, which can exist if certain other considerations are taken. For more about this, see Chapter 3: Designing and implementing an on-premises database infrastructure. Otherwise, do not store any non-temporary files in the temporary storage.

The following sample scenario is a good starting point for a volume layout for your SQL Server installation (the volume letters don’t matter):

- **Volume C.** OS, some SQL Server files must install here.
- **Volume E.** SQL Server installation files, log files, SQL Server database data files.
- **Volume F.** SQL Server database log files.
- **Volume G.** SQL Server TempDB data files and log file. (Alternatively, use the D: Temporary Storage volume on Azure Windows VMs.)
- **Volume H.** SQL Server backups (if written locally.)

And, here are some more advanced volume decisions:

- Use additional volumes for your largest data files (larger than 2 TB) for storage manageability:
 - For the most active databases
 - For FILESTREAM filegroups
 - For database replication snapshot files
 - For the Windows Page File, especially for servers with large amounts of memory

Inside OUT

Why separate SQL Server files onto different volumes?

There are good reasons to separate your SQL Server files onto various volumes, and not all of them are related to performance. You should still separate your files onto different volumes even if you exclusively use a Storage-Area Network (SAN).

More discrete storage I/O on a physical server with dedicated drives means better performance. But even in a SAN, separating files onto different volumes is also done for stability. Think of the volumes as bulkheads on a submarine. If a volume fills and has no available space, files cannot be allocated additional space. On the OS volume, running out of free space would result in Windows Server stability issues, user profile and remote desktop problems at least, and impact to other applications.

Important SQL Server volume settings

There are some settings to consider for volumes that host SQL Server data and log files, and this guidance applies specifically to these volumes. For other volumes—for example, those that contain the OS, application files, or backup files—the default Windows settings are acceptable unless otherwise specified.

When adding these volumes to Windows, there are four important volume configuration settings required to examine or discuss with your storage administrator.

- When creating new drives, opt for GUID Partition Table (GPT) over Master Boot Record (MBR) disk types for new SQL Server installations. GPT is a newer disk partitioning scheme than MBR, and GPT disk support files and volumes larger than 2 TB, whereas the older MBR disk type is capped at 2 TB.
- The appropriate file unit allocation size for SQL Server volumes is 64 KB, with few exceptions. Setting this to 64 KB for each volume can have a significant impact on storage efficiency and performance. The Windows default is 4 KB, which is not optimal for SQL Server data and log files.

To check the file unit allocation size for an NT File System (NTFS) volume, run the following from the Administrator: Prompt, repeating for each volume:

```
fsutil fsinfo ntfsinfo d:
```

The file unit allocation size is returned with the Bytes Per Cluster; thus the desired 64 KB would be displayed as 65,536 (bytes). If formatted as the default, this will display 4096. Correcting the file unit allocation size requires formatting the drive, so it is important to check this setting prior to installation.

If you notice this on an existing SQL Server instance, your likely resolution steps are to create a new volume with the proper file unit allocation size and then move files to the new volume during an outage. Do *not* format or re-create the partition on volumes with existing data: you will of course lose the data.

- Modern storage devices are currently in a transition between disks that use a Bytes per Physical Sector size of 512 bytes (the old standard) and “4K Native” disks that have both a Bytes per Sector size and a Bytes per Physical Sector size of 4 KB. Usually a DBA will not notice or even be aware of this difference. When configuring Availability Groups or log shipping between servers on different storage systems with mixed Bytes per Physical Sector modes, however, this can result in very poor performance, with the transaction logs unable to truncate, and the error message “There have been *nnn* misaligned log IOs which required falling back to synchronous IO.” You may encounter this with hybrid Availability Groups spanning on-premises and Azure VM-based SQL Server instances, for example.

This cannot be resolved via a formatting decision, but can potentially be resolved via hardware-level storage or firmware settings. To avoid this, all storage that hosts the transaction log files of SQL Servers in an Availability Group or log shipping relationship should have the same Bytes per Physical Sector.

A workaround is to apply Trace Flag 1800 as a startup flag on the SQL Server instances that use storage without having a Bytes per Physical Sector setting of 4K. TF1800 overrides disk default behavior and writes the transaction log in 4 KB sectors, resolving the issue. TF1800 must be enabled on the on-premises SQL Server instances, in the case of using the older on-premises and Azure VM Availability Group.

Check the Bytes per Physical Sector setting of a volume by using the same Fsutil command noted in the previous code sample.

- There is a hardware-level concept related to file unit allocation size called “disk starting offset” that deals with how Windows, storage, disk controllers, and cache segments align their boundaries. Aligning disk starting offset was far more important prior to Windows Server 2008. Since then, the default partition offset of 1,024 KB has been sufficient to align with the underlying disk’s stripe unit size, which is a vendor-determined value, and rarely a concern for DBA’s. Still, it should be verified upon first use of a new storage system or the migration of disks to a new storage system. This can be verified in consultation with the drive vendor’s information.

To access the disk starting offset information, run the following from the Administrator: Prompt:

```
wmic partition get BlockSize, StartingOffset, Name, Index
```

A 1024 KB starting offset is a Windows default, which is displayed as 1048576 (bytes) for Disk #0 Partition #0.

Similar to the file unit allocation size, the only way to change a disk partition's starting offset is destructive—you must re-create the partition and reformat the volume to align with the vendor-supplied offset.

SQL Server editions

The following are brief descriptions for all of the editions in the SQL Server family, including past editions that you might recognize. It's important to use the appropriate licenses for SQL Server even in preproduction systems.

NOTE

This book is not intended to be a reference for licensing or sales-related documentation; rather, editions are a key piece of knowledge for SQL administrators to understand what features may or may not be available.

- **Enterprise edition.** Appropriate for production environments. Not appropriate for preproduction environments such as User Acceptance Testing (UAT), Quality Assurance (QA), testing, development, or a sandbox. For these environments, instead use the free Developer edition. You'll have a far easier time in a licensing audit if your pre-production environment installations are Developer edition.
- **Developer edition.** Appropriate for all preproduction environments, especially those under a production Enterprise edition. Not allowed for production environments. This edition supports the same features and capacity as Enterprise edition and is free.
- **Standard edition.** Appropriate for production environments. Lacks the scale and compliance features of Enterprise edition that are required in some regulatory environments. Limited to the lesser of 4 sockets or 24 cores and also 128 GB of buffer pool memory, whereas Enterprise edition is limited only by the OS for compute and memory.
- **Web edition.** Appropriate for production environments but limited to low-cost server environments for web applications.
- **Express edition.** Not appropriate for most production environments or preproduction environments. Appropriate only for environments in which data size is small, is not expected to grow, and can be backed up with external tools or scripts (because Express edition has no SQL Server Agent to automate backups). The free Express edition is ideal for production proof-of-concepts, lightweight applications, or student projects. It lacks some critical features and is severely limited on compute (lesser of 1 socket or 4 cores), available buffer pool memory (1,410 MB), and individual database size (10 GB cap).
- **Express with Advanced Services.** Similar to Express edition in all caveats and limitations, this edition includes some additional features including R integration, full-text search, and distributed replay.

- **Evaluation Edition.** Functionally the same as Enterprise edition, and free with a 180-day shutdown timer, but it isn't supported. Can be upgraded to any edition but Express. Do not use it if you plan for a clustered installation, because an upgrade in that case is not supported.
- **It's worth noting that the hardware limitations of SQL Server editions have not changed since SQL Server 2016.**

NOTE

When you run the SQL Server 2019 installer, you are prompted to install a number of features outside of the core database features. Installing SQL Server features on multiple Windows servers requires multiple licenses per server, even if you intend to install each SQL Server instance's features only once.

There is an exception to this rule, and that is if you have licensed all physical cores on a virtual host server for SQL Server Enterprise edition, and purchased Software Assurance, you can install any number or combination of SQL Server instances and their standalone features on virtual guests.

Changing SQL Server editions and versions

Upgrading editions in-place is supported by a feature of the SQL Server 2019 installer. You can upgrade in the following order: Express, Web, Standard, and Enterprise.

It is important to note that you cannot downgrade a SQL Server version or licensed edition. This type of change requires a fresh installation and migration. For example, you cannot downgrade in-place from SQL Server 2019, from Enterprise edition to Standard edition.

In-place upgrades for major versions (from 2017 to 2019, for example) is supported but not recommended. Instead, we strongly recommend that you perform a fresh installation of the newer version and then migrate from old to new instances. This method offers major advantages in terms of duration of the planned outage, rollback capability, and robust testing in parallel.

Although in-place upgrades to SQL Server 2019 are not recommended, they are supported for versions as old as SQL Server 2012 on Windows Server 2016. SQL Server 2019 is the first SQL Server version that does not support Windows Server 2012 or 2012 R2.

A supported upgrade also assumes that the operating system and previous version of SQL Server are not 32-bit installations. Beginning with SQL Server 2016, SQL Server is available only for 64-bit platforms. For more information on upgrades to SQL Server 2019, visit <https://docs.microsoft.com/sql/database-engine/install-windows/supported-version-and-edition-upgrades-version-15>.

Beginning with SQL Server 2019, a SQL Server upgrade from SQL Server 2008 is no longer supported, though individual databases can still be migrated to SQL Server 2019. You can attach or restore its databases to SQL Server 2019, although they will be upgraded to compatibility level 100, the version level for SQL 2008.

Installing a new instance

In this section, you learn how to begin a new SQL Server 2019 instance installation, upgrade an existing installation, or add features to an existing instance.

The instructions in this chapter are the same for the first installation or any subsequent installations, whether it is for the default or any named instances of SQL Server 2019. As opposed to an exhaustive step-by-step instruction list for installations, we've opted to cover the important decision points and the information you need and highlight new features from SQL Server 2019.

It's important to note that even though you can change *almost* all of the decisions you make in SQL Server Setup after installation, those changes potentially require an outage or server restart. Making the proper decisions at installation time is the best way to ensure the least administrative effort. Some security and service account decisions should be changed only via the SQL Server Configuration Manager application, not through the Services console (services.msc). This guidance will be repeated elsewhere for emphasis.

We begin by going through the typical interactive installation. Later in this chapter, we go over some of the command-line installation methods that you can use to automate the installation of a SQL Server instance.

Planning for multiple SQL Server instances

You can install as many as 50 SQL Server instances on a Windows Server; obviously, we do not recommend this. In a Windows failover cluster, the maximum number of SQL Server instances is reduced by half if you're using shared cluster drives.

Only one of the SQL Server instances on a server can be the default instance. All, or all but one, of the SQL Server instances on a SQL Server will be named instances. The default instance is reachable by connecting to the name of the Windows Server, whereas named instances require an instance name, for example, *Servername\InstanceName*. The SQL Browser service is required to handle traffic for named instances on the SQL Server.

For example, you can reach the default instance of a SQL Server by connecting to *servername*. All named instances have a unique instance name, such as *servername\instancename*.

Creating an Azure VM running SQL Server

You should not need to install SQL Server on new Azure VMs, because provisioning new VMs with various versions and editions of Windows Server and SQL Server are available in the Azure Marketplace. For more information about bringing your existing SQL Server licensing to Azure VMs, visit <https://azure.microsoft.com/blog/easily-bring-your-sql-server-licenses-to-azure-vm/>.

When manually creating a new VM with SQL Server, however, there are still some configuration choices you can make in the browser. We will walk through some of those decision points in this section, although much of this is already handled for you.

Specifically in the *SQL Server settings* page of the *Create a virtual machine* wizard, you can choose to enable *Mixed Mode Authentication*. If you do, you must specify a username and password, but note that this probably isn't needed for you—administrators can and should use Azure Active Directory authentication when possible with your Azure VMs.

You can configure the Azure VM to be automatically powered down, which is very handy for rarely used VMs in your own sandbox. You can use Azure Automation to easily schedule VMs to run during pre-scheduled hours, thus lowering the monthly cost.

You can also enable Azure-specific integrations with Azure Key Vault, automatic patching, and automatic SQL backups for your new VM from the Azure Portal. Use Tags to organize your Azure infrastructure with information on environments, applications, locations, or other custom departmental metadata as desired, making it easy to monitor and manage objects in groups.

In the *Review + create* page, you can review the expected hourly spend and all of your Azure VM settings. Don't miss this detail, which is similar to SQL Server setup: there is a link to download a template for automation. An Azure VM template is a .json file that stores the options needed to create Azure resources, so that you can reference it within the Azure Portal, PowerShell, the Azure CLI (-Line Interface) and others to create Azure VMs with the same choices. Use the .json file to create a new Azure VM, for example, by choosing "Create a Resource" in Azure, then "Template deployment (deploy using custom templates)".

- You'll find much more on Azure infrastructure, including the differences between Azure VMs, Azure SQL Database, and Azure SQL Database Managed Instances in Chapter 16 "Designing and implementing hybrid and Azure database infrastructure."

Inside OUT

How does licensing work in Azure VMs?

This isn't an authoritative book on Azure or SQL Server licensing, but speaking broadly, there are two types of SQL Server licensing agreements for Azure VMs:

SQL Server VM images in the Azure Marketplace contain the SQL Server licensing costs as an all-in-one billing package.

Alternatively, if you'd like to leverage your existing Enterprise licensing agreement using the Azure Hybrid Benefit, there are three options:

- Bring-your-own-license (BYOL) VM images available for you to provision using the same process and then later associate your existing Enterprise license agreements. The image names you're looking for here are prefixed with BYOL.
- Manually upload an .iso to the VM and install SQL Server 2019 as you would on any other Windows Server.
- Upload an image of an on-premises VM to provision the new Azure VM.

It is important for you to keep in mind that you cannot change from the built-in licensing model to the BYOL licensing model after the VM has been provisioned. You need to make this decision prior to creating your Azure VM.

Installing SQL Server on Windows

The rest of this chapter is dedicated to installations of SQL Server not part of a pre-made Azure Marketplace VM and apply to installation of SQL Server on any Windows Server.

While logged in as a local Windows administrator, begin by mounting the installation .iso to the Windows server. These days, this rarely involves inserting a physical disc or USB flash drive; although you can use them if necessary.

Launching SQL Server Setup

You should not run Setup with the installation media mounted over a remote network connection, via a shared remote desktop drive, or any other high-latency connection. For a faster SQL Server Setup experience, unpack the contents of the .iso file to a physical file folder local to the server, instead of mounting a file or launching Setup from a location over the network.

Start setup.exe on the SQL Server Setup media, running the program as a Windows user with administrator privileges. If AutoPlay is not turned off (it usually is), Setup.exe will start when you

first mount the media or double-click to open the .iso. Instead, as a best practice, right-click Setup.exe and then, on the shortcut menu that appears, click **Run As Administrator**.

We'll review here a few items (not all) in the SQL Server Installation Center worth noting before you begin an installation.

In the tab pane on the left, click **Planning** to open a long list of links to Microsoft documentation websites. Most helpful here might be a standalone version of the System Configuration Checker, which you run during SQL Server Setup later, but it could save you a few steps if you review it now. A link to download the Data Migration Assistant (DMA) is also present, which is a helpful Microsoft-provided tool when upgrading from prior versions of SQL Server.

On the *Maintenance* page, you will find the following:

- A link to launch the relatively painless Edition Upgrade Wizard. This is only for promoting your existing installation's edition as we discussed earlier.
 - The Repair feature is not a commonly used feature. Its use is necessitated by a SQL Server with a corrupted installation. You might also need to repair an instance of SQL Server when the executables, .dll files, or Registry entries have become corrupted or damaged by disk corruption, antivirus, or malicious activity. A failed SQL Server in-place upgrade or cumulative update installation might also require a Repair, which could be better than starting from scratch.
 - Note that whereas removing a node from an existing SQL Server failover cluster is an option in the *Maintenance* page, adding a node to an existing SQL Server failover cluster is an option in the *Installation* page.
 - On the *Advanced* page, there is a link to perform an installation based on a configuration file. We will discuss how to easily generate and use a configuration file later in this chapter, in the section *Automating SQL Server Setup by using configuration files*. If you are tasked with installing multiple SQL Servers with mostly common settings, consider this time-saving method. There are also links to wizards for advanced failover cluster installations.
- We discuss Failover Cluster Instances (FCIs) in Chapter 11, "Implementing high availability and disaster recovery."

Windows Update in the SQL Server Setup

Since SQL Server 2012, the SQL Server installer has had the ability to patch itself while within the Setup wizard. The *Product Updates* page is presented after the *License Terms* page, and, after you accept it, it is downloaded from Windows Update (or Windows Server Update Services) and installed along with other SQL Server Setup files.

This is recommended, and so a SQL Server 2019 Setup with Internet connectivity is the easiest way to carry out the installation. This also could be described as a way to “slip-stream” updates, including hotfixes and cumulative updates, into the SQL Server installation process, eliminating these efforts post-installation.

For servers without Internet access, there are two Setup.exe parameters that support downloading these files to an accessible location and making them available to Setup. When starting Setup.exe from Windows PowerShell or the command line (you can read more about this in the next section), you should set the `/UpdateEnabled` parameter to `FALSE` to turn off the download from Windows Update. The `/UpdateSource` parameter can then be provided as an installation location of .exe files. Note that the `/UpdateSource` parameter is a folder location, not a file. You will find more on these two parameters later in the “Installing by using a configuration file” section.

Regardless, after installation is complete and before the SQL Server enters further use in your team, verify that the latest SQL Server patches have been applied. For SQL Server 2019, see the official build versions site:

<https://support.microsoft.com/help/4518398/sql-server-2019-build-versions>.

Installing SQL Server *stand-alone installation*

Although what follows in this chapter is not a step-by-step walk-through, we’ll cover key new features and decision points of the *New SQL Server stand-alone installation* option of the SQL Server Installation Center.

Inside OUT

Where is SQL Server Management Studio and SQL Server Data Tools?

SQL Server Management Studio, SQL Server Data Tools (for Visual Studio 2015 and higher), and SQL Server Reporting Services are no longer installed with SQL Server’s traditional setup media. These products are now updated regularly (as often as monthly) and available for download.

You should keep up-to-date versions of SQL Server Management Studio (SSMS) on administrator workstations and laptops.

Avoid installing SSMS locally on the SQL Server if possible. In fact, avoid needing to use Remote Desktop Connection to manage and administer the SQL Server altogether. For all SQL Server platforms, try to use SSMS, Azure Data Studio, PowerShell, and other tools to do as much of your work on SQL Server remotely as possible.

Granting Perform Volume Maintenance Tasks

On the same Server Configuration page on which service accounts are set, you will see a check box labeled *Grant Perform Volume Maintenance Task privilege to the SQL Server Database Engine Service*. This option was added to SQL Server Setup in SQL Server 2016.

This automates what used to be a standard post-installation checklist step for SQL DBAs since Windows Server 2003. The reason to grant this permission to use instant file initialization is to speed the allocation of large database data files, which could dramatically reduce the Recovery Time Objective (RTO) capacity for disaster recovery.

This can mean the difference between hours and minutes when restoring a very large database. It can also have a positive impact when creating databases with large initial sizes, or in large autogrowth events; for example, with multiple data files in the TempDB (more on this next). It is recommended that you allow SQL Server Setup to turn on this setting.

Inside OUT

How can you verify that Instant File Initialization is enabled?

IFI is granted to SQL Server service account via the *Perform Volume Maintenance Tasks* permission in Local Security Policy on the Windows server. But it's actually quite easy to verify whether or not IFI is in place for the SQL Server service via the `sys.dm_server_services` dynamic management view. The column `instant_file_initialization_enabled` was first added in a series of patches to supported SQL Servers: in SQL Server 2012 SP4, SQL Server 2014 SP2, SQL Server 2016 SP1.

```
SELECT servicename, instant_file_initialization_enabled
FROM sys.dm_server_services
WHERE filename LIKE '%sqlservr.exe%';
```

- For more information on instant file initialization, see Chapter 3.

Instance Collation

The Collation tab on the Server Configuration page allows you to choose a collation for the Database Engine. The collation determines how character data is stored, sorted and compared. For more information, see Chapter 7: "Understanding table features" in the section on Collation.

Initially the instance collation provided in SQL Setup is the default collation for the server's regionalization, but you may need to change this collation based on vendor or developer specifications.

While changing the collation of a database is easy, the instance collation is important to get right at the time of SQL Server installation, as changing the instance collation is quite difficult. To change the collation of the SQL Server instance, reference this lengthy and difficult Microsoft guide: <https://docs.microsoft.com/sql/relational-databases/collations/set-or-change-the-server-collation>. Note that in the case of Azure SQL managed instances, you cannot change the server-level collation after it is created. For more information, visit: <https://docs.microsoft.com/sql/relational-databases/collations/set-or-change-the-server-collation#setting-the-server-collation-in-managed-instance>.

The server collation you set here acts as the collation for all system databases as well as the default for any newly created user databases. In SQL Server 2019 and for new application development, you may choose to take advantage of the new support for UTF-8 collations as the server default. Pictured below is the new interface in SQL Server Setup for choosing a UTF-8 collation.

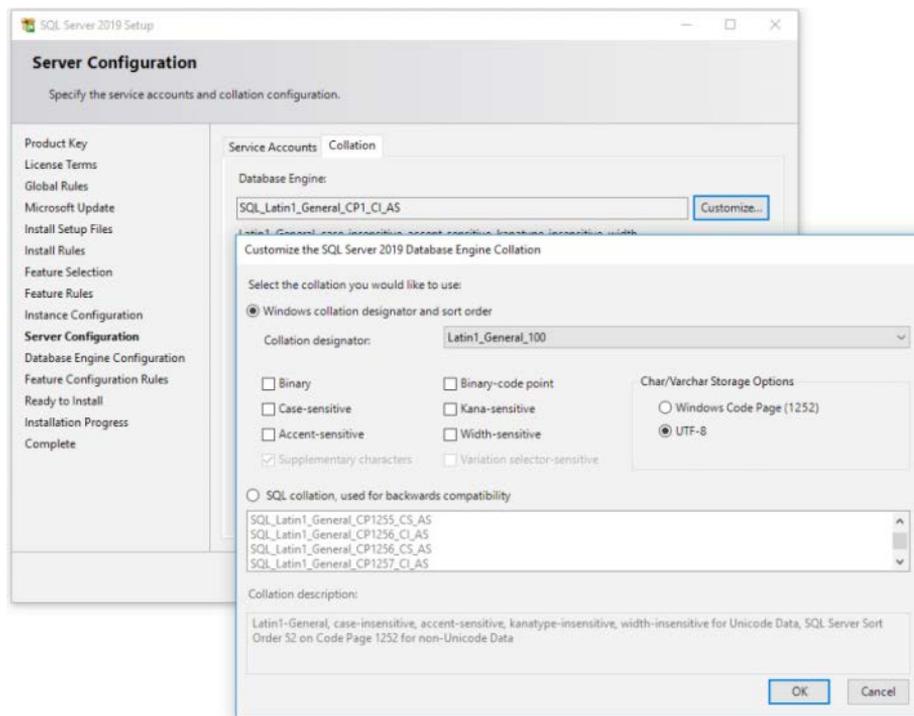


Figure 4-1 The options to set a server-level Windows collation using the new UTF-8 collation options now available in SQL Server 2019 Setup.

Mixed Mode authentication

SQL Server supports two modes of authentication: Windows and SQL Authentication. One is preferable to the other, and in multiple places in this book we will emphasize the preference for Windows Authentication over SQL Authentication.

- ▶ You can read more on this topic in Chapter 12, “Administering security and permissions,” but it is important to note this decision point here.

Ideally, all authentication is made via Windows Authentication, through types of server principals called *logins*, that reference Windows accounts, which are ideally Active Directory domain accounts. These are created by your existing enterprise security team, which manages password policy, password resets, password expiration, and so on.

A redundant security model for connecting to SQL Server also exists within each instance: SQL Server Authenticated logins. Logins are maintained at the SQL Server level, are subject to local policy password complexity requirements, are reset/unlocked by SQL DBAs, have their own password change policy, and so forth.

Enabling Mixed Mode (SQL and Windows Authentication Mode) activates SQL Authenticated logins. It is important to note that SQL Authentication is not on by default, and isn't the recommended method of connection. By default, the recommended Windows Authentication is turned on and cannot be turned off. When possible, applications and users should use Windows Authentication.

Enabling Mixed Mode also activates the “sa” account, which is a special built-in SQL Server Authentication that is a member of the server sysadmin role. Setup will ask for a strong password to be provided at this time.

- ▶ You can learn more about the “sa” account and server roles in Chapter 7.

If you find you have an actual need to enable SQL Server Authentication, but didn't do this during SQL Server Setup, you can do this later on by connecting to the SQL Server instance via Object Explorer in SQL Server Management Studio. To do so, right-click the server name and then, on the shortcut menu that opens, click **Properties**, and then click the **Security** page, change to Mixed Mode. You must perform a SQL Server service restart to make this change effective.

Default settings for the TempDB database

Starting with SQL Server 2016, SQL Server Setup provides a more realistic default configuration for the number and size of TempDB data files. This has been a common to-do list for all post-installation checklists for DBAs since the early days of SQL Server.

The TempDB database page in SQL Server Setup provides not only the ability to specify the number and location of the TempDB's data and log files, but also their initial size and auto-growth rates. The best number of TempDB data files is almost certainly greater than one, and less than or equal to the number of logical processor cores, including hyperthreading for local machines. For example, with 16 logical processors, SQL Server Setup will default the installation to have 8 TempDB data files.

Adding too many TempDB data files can degrade SQL Server performance, perhaps severely if you add too many. For example, with 20 logical processors, SQL Server Setup will still default the installation to have 8 TempDB data files. If you add 20 TempDB data files, SQL Server may struggle to respond.

- For more information on the best number of TempDB data files, see Chapter 3.

Specifying TempDB's initial size to a larger, normal operating size is important and can improve performance after a SQL Server restart when the TempDB data files are reset to their initial size. Setup accommodates an individual TempDB data file initial size up to 256 GB. For data file initial sizes larger than 1 GB, you will be warned that SQL Server Setup can take a long time to complete if instant file initialization is not turned on.

Since SQL Server 2016, all TempDB files autogrow at the same time, keeping file sizes the same over time, which is critical to the way multiple TempDB data files are used. This is superior to the old way of ensuring TempDB data files stay the same size, using the server-level setting via server Trace Flag 1117, which applied the data file growth behavior to all databases. Trace Flag 1117 is no longer necessary.

Note also the new naming convention for the second TempDB data file and beyond: *tempdb_mssql_n.ndf*. A SQL Server uninstallation will automatically clean up TempDB data files with this naming convention—for this reason, we recommend that you follow this naming convention for TempDB data files.

- TempDB is discussed in greater detail in Chapter 3.

Default settings for MaxDOP

New in SQL Server 2019 are defaults for the configuration of the server-wide Maximum Degrees of Parallelism setting, included on the *Database Engine Configuration* page under the new *MaxDOP* tab.

Similar to how new TempDB defaults since SQL Server 2016 are dependent on the detected processors, a suggested default MaxDOP is also configured based on the number of logical processors. For many servers with 16 or fewer virtual processor cores, the default is the same as

the number of the cores, effectively the same as *MaxDOP* setting = 0, which allows for unlimited parallelism.

For example, with 8 logical processors, SQL Server Setup will default the installation to use MaxDOP 8, as pictured in Figure 4-2. With over 16 logical processors, SQL Server Setup may default to half the number of logical processors—at most 16. For example, with 20 logical processors, SQL Server Setup will default the installation to use MaxDOP 10. For more recommendations about MAXDOP, visit Microsoft Support at <https://support.microsoft.com/help/2806535>. See also the section on Max degree of parallelism in Chapter 3.

You can always reconfigure the MaxDOP after installation, without a restart, though not without potential disruption. While changing the server-wide (or database-level) MaxDOP setting takes effect immediately, it is definitely not advisable to do so during normal production operating hours, because it can lead to widespread plan recompilation and a heavy CPU spike. This server-wide MaxDOP setting can be overridden at the database, query, or Resource Governor group level.

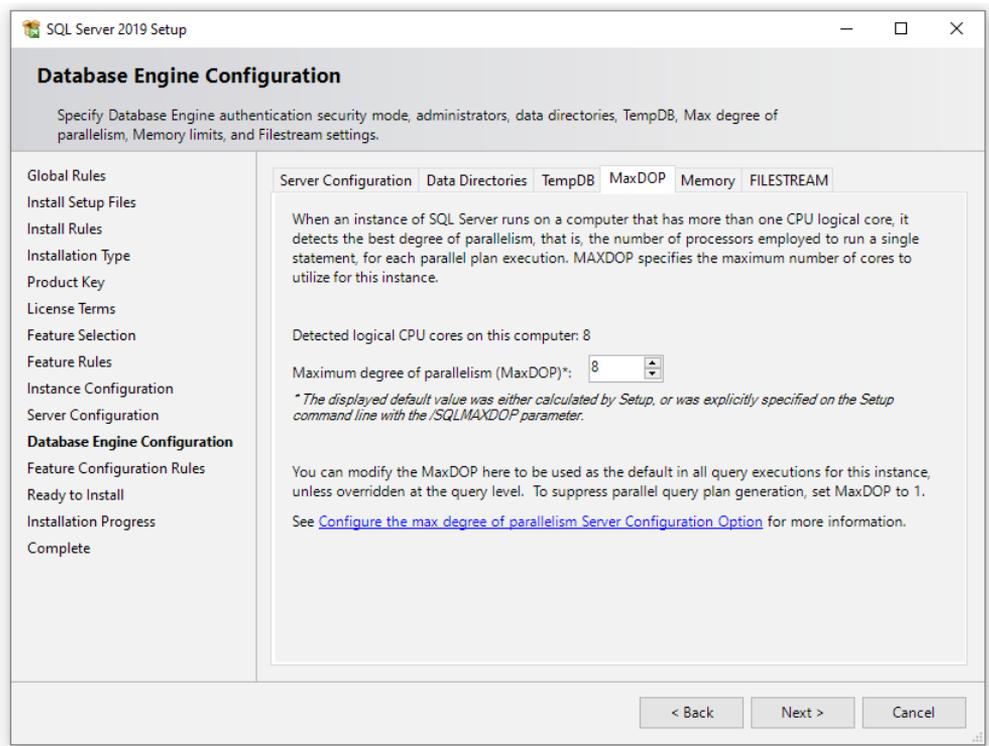


Figure 4-2 The new MaxDOP tab in the Database Engine Configuration tab in SQL Server 2019 Setup, which has recommended a MaxDOP of 8 for a server with 8 virtual cores. This is effectively the same as MaxDOP 0, but offers the administrator an option to potentially change MaxDOP at the time of installation.

NOTE

Some applications, including Microsoft SharePoint and other third-party vendor applications, recommend that parallelism be disabled on their databases. This can be accomplished at the server level now, or, can be configured and overridden at each database level after SQL Server Setup is complete. Consult your vendor's specifications and recommendations documentation.

- For much more information on performance tuning, parallelism, and the MaxDOP setting, see Chapter 14, "Performance tuning SQL Server."

Default settings for Maximum Server Memory

New in SQL Server 2019 are defaults for the configuration of the instance-level Maximum Server Memory option, a common post-installation checklist item, under the Memory tab of the Database Engine Configuration page. SQL Server Setup makes a guess based on total server memory for an appropriate option. In previous versions of SQL Server, it was important to remember to go and change the Max Server Memory setting after installation was complete, otherwise SQL Server memory would be uncapped, and have access to all memory on the server.

Now in SQL Server 2019, this max server memory option can be configured intelligently at the time of installation. It's important to note (and there's a checkbox to accept this guess) that SQL Server Setup is assuming this SQL Server instance will run alone on this server. If you expect to host other applications on this server, or memory-heavy features of SQL Server on the same server, such as SSAS or SSRS, you should reduce the maximum server memory for the SQL Server instance further.

- We discussed the Maximum Server Memory setting in Chapter 3, in the "Configuration Settings" section.

An example of the new max server memory recommendation configuring a Windows Server with one SQL Server instance and 16 GB of memory, SQL Server setup recommends a Max Server Memory setting of 12672 MB, as seen in Figure 4-3.

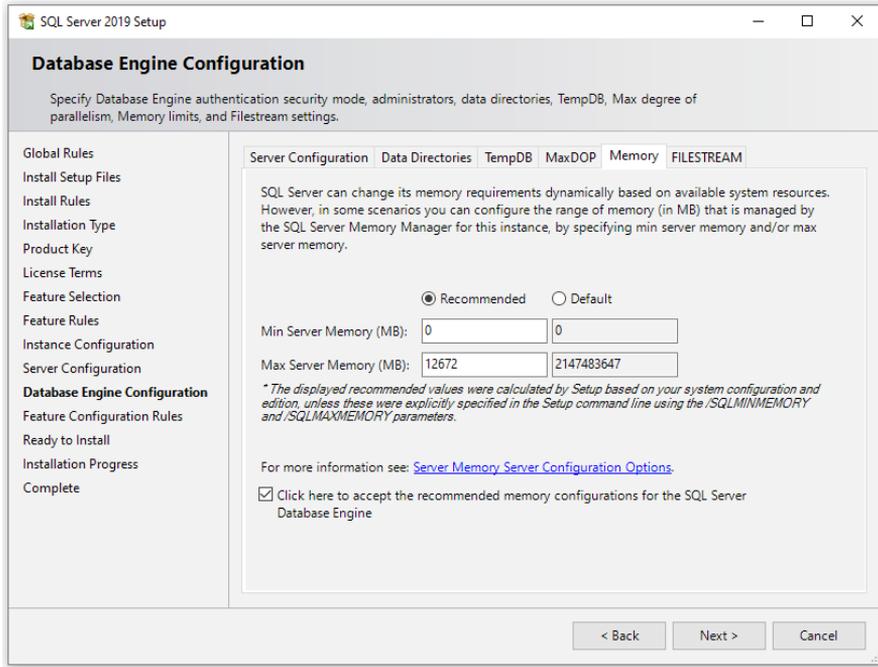


Figure 4-3 The new Memory tab in the Database Engine Configuration tab in SQL Server 2019 Setup.

Just above the *Maximum Server Memory* setting is the *Minimum Server Memory* setting, which establishes a floor for memory allocation. It is generally unnecessary to change this setting from the default of 0. You might find this setting useful for situations in which the total system memory is insufficient and many applications, including SQL Server instances, are present. The minimum server memory is not immediately allocated to the SQL Server instance upon startup; instead, it does not allow memory below this level to be freed for other applications.

After installation, server memory settings are accessible via SQL Server Management Studio, in Object Explorer, and on the Server Properties page.

You should assure that SQL Server leaves enough memory for the operating system and other applications. Keep in mind that SQL Server will slowly consume more memory over time, and may take hours or days, depending on your business cycle, for the SQL Server instance to consume the maximum amount of memory made available. Similarly, lowering this setting after installation and during operation does not return SQL Server memory back to the OS immediately; rather, it does so over time during SQL Server activity. Similarly, increasing this setting will not immediately show the effect of a change in memory utilization.

Installing common features

Aside from the SQL Server service itself, other features of the product might be common to your installations. SQL Server Analysis Services, SQL Server Integration Services, and SQL Server Reporting Services are part of the license and are provided at no additional cost. If you need them, this section covers installing these features using Setup. Later in this chapter, we cover the post-installation steps necessary to use them.

Installing SQL Server Analysis Services

Installing SQL Server Analysis Services (SSAS) requires you to make a decision at installation time regarding the mode in which it can be installed. Each instance of SQL Server Analysis Services can be in only one mode, which means that with a single license, you can run only Multidimensional mode, the newer Tabular mode (introduced in SQL 2012), or the Power Pivot mode.

Ask your business intelligence decision makers which platform you should use, though for most new development, Tabular mode is popular and recommended. Tabular mode databases can also run in Azure Analysis Services. The following are brief descriptions of each mode:

- **Multidimensional mode.** This is the SSAS setup that was first introduced in SQL 2000. This is also the only mode to support data mining and other features that existing SSAS data models predating SQL Server 2012 may be dependent on. The primary language for building and querying multidimensional models is MDX.
 - **Tabular mode.** This is the newer and recommended SQL Server Analysis Services setup that was first introduced in SQL 2012, using the in-memory VertiPaq processing engine. Since SQL Server 2017, this is the default installation mode selected on the Analysis Services Configuration page of Setup. The primary language for building and querying tabular models is DAX, which is similar to the Excel function language.
 - **Power Pivot mode.** This mode installs SQL Server Analysis Services in the Power Pivot for SharePoint mode. Power Pivot workbooks use both DAX and MAX. Note that Analysis Services Power Pivot for SharePoint support for Microsoft SharePoint 2019 has been discontinued.

For more on the differences between these SSAS installation options, visit <https://docs.microsoft.com/analysis-services/comparing-tabular-and-multidimensional-solutions-ssas>.

Inside OUT

What if you choose the wrong SQL Server Analysis Services mode?

If you choose one SQL Server Analysis Services mode at installation but your business intelligence developers want another mode, the supported option is to uninstall and

reinstall the SQL Server Analysis Services feature. Changing the SQL Server Analysis Services mode from Multidimensional to Tabular, or vice versa, after installation is not supported, and administrators are specifically warned not to do this.

Packages developed for each mode are not supported for the other. If no databases have been deployed to the SQL Server Analysis Services server instance, changing the *DeploymentMode* property in the MSMDSRV.ini file should make it possible to change an existing instance, but, again, this is not a supported change. The file is located in *%Programfiles%\Microsoft SQL Server\MSAS15.instance\OLAP\Config*.

Installing SQL Server Integration Services

The SQL Server Integration Services instance for SQL Server 2019 is installed once per server per version, not once per instance, like other features. Starting in SQL Server 2017, however, a new Integration Services Scale Out Configuration is available. We discuss this new feature further in the next section.

A 64-bit version of SQL Server Integration Services is installed on 64-bit operating systems. If you worry about connecting to 32-bit servers, data sources, or applications installations (such as Microsoft Office), don't—those connections are not dependent on the 32-bit/64-bit installation and are handled at the package or connection-string level. Unlike other features, you can install SQL Server Integration Services on a 32-bit OS; however, we do not recommend this.

Installations of different versions of SQL Server Integration Services are installed side-by-side on a server; specifically, the service SQL Server Integration Services 15.0 is compatible with prior versions.

Outside of configuring the service account, you do not need any additional configuration when installing SQL Server Integration Services during SQL Server Setup. The default virtual service account is *NT Service\MsDtsServer150*.

Inside OUT

Should you install SQL Server Integration Services (SSIS) alone on a server?

A standalone installation of SQL Server Integration Services without a matching SQL Server Database Engine is possible but not recommended. For the modern Project Deployment model of SQL Server Integration Services, the storage and logging of packages will still be dependent on a SQL Server Database Engine, and the execution of packages on a schedule still requires a SQL Agent service. So isolation of the SQL Server Integration Services workload is not best isolated in this way. A dedicated installation

including the SQL Server Database Engine and SQL Server Agent is a better configuration to isolate SQL Server Integration Services package runtime workloads from other database workloads. Both of these options would carry the same licensing cost.

Installing SQL Server Integration Services Scale Out configuration

Integration Services supports a Scale Out configuration by which you can run a package on the same or multiple SQL Server instances, a new feature that was first introduced in SQL Server 2017. This also allows for high availability of SQL Server Integration Services, and a similar architecture allows for integration and “lift and shift” code deployments from on-premises SQL Server Integration Servers to the Azure Integration Runtime.

The master node talks to worker nodes in a SQL Server Integration Services Scale Out, with the communication over a port (8391 by default) and secured via a new Secure Sockets Layer (SSL) certificate. The SQL Server installer can automatically create a 10-year self-signed certificate and endpoint for communication at the time the master node is set up.

When adding another SQL Server Integration Services installation as a Scale Out Worker, start the new SQL Server Integration Services Manage Scale Out window via SQL Server Management Studio. Right-click the Catalog you have created, and then click **Manage Scale Out**. At the bottom of the page, click the + button to add a new Scale Out Worker node. Provide the server name on which to connect. If using a named instance, provide only the server name of the node; do not include the instance name. A dialog box confirms the steps taken to add the Worker node, including copying and installing certificates between the Worker node and Master node, updating the endpoint and `HttpsCertThumbprint` of the worker, and restarting the Worker’s Scale Out service. After the worker node is added, refresh the Worker Manager page, and then click the new Worker node entry, which will be red. You must turn on the Worker Node by clicking **Enable Worker**.

You also can copy and install the certificates manually between servers. You will find them in: `%program files%\Microsoft SQL Server\150\DTS\Binn\`. For more information on certificates between servers, visit <https://docs.microsoft.com/sql/integration-services/scale-out/deal-with-certificates-in-ssis-scale-out>. For a Microsoft-provided walkthrough of setting this up, visit <https://docs.microsoft.com/sql/integration-services/scale-out/walkthrough-set-up-integration-services-scale-out>.

One major security difference with Scale Out is that even though the SQL Server Integration Services Service Account doesn’t run packages or need permission to do very much, the Scale Out Master and Worker service accounts actually do run packages. The SQL Server Integration Services service account is different from the Scale Out Master and Scale Out Worker service accounts.

The Worker and Master nodes do not appear in SQL Server Configuration Manager (as of SQL 2019) but do appear in Services.msc. By default, these services run under virtual accounts *NT Service\SSISScaleOutMaster150* and *NT Service\SSISScaleOutWorker150*, but you might want to change these to a Windows-authenticated Domain service account that will be used to run packages across the Scale Out.

Installing SQL Server Reporting Services

Starting with SQL Server 2017, SQL Server Reporting Services is no longer found in the SQL Server Setup media; it is instead available as a simplified, unified installer and a small download. SQL Server Reporting Services is now a 95+MB download named *SQLServerReportingServices.exe*, but still needs a SQL Server Database Engine instance as part of the license to host the two Report Server databases.

Note that SQL Server Reporting Services isn't free, and that the separate installer isn't a licensing change, though SQL Server Express with Advanced Services offers some limited SSRS support. For more information on the limitations of SSRS with SQL Server Express license, view <https://docs.microsoft.com/sql/reporting-services/reporting-services-features-supported-by-the-editions-of-sql-server-2016>.

To install SSRS, you will need to provide a license key upon installation in a production environment. You can choose a free edition to install (Evaluation, Developer, or Express), though you should note that Developer edition is not allowed in a production environment.

The "native" mode of SQL Server Reporting Services is now the only mode, since SQL Server 2017. If you are familiar with Reporting Services Report Manager in the past, accessible via the URL *servername/Reports*, that is the "native mode" installation of Reporting Services.

You'll notice the Report Server Configuration Manager in a new location, in its own Program Files menu: *Microsoft SQL Server Reporting Services*. After installation, start the Report Server Configuration Manager (typically installed in a path like *\Program Files (x86)\Microsoft SQL Server\150\Tools\Binn\RSConfigTool.exe*). The Report Server Configuration Manager application itself is largely unchanged since SQL 2008.

The virtual service account *NT SERVICE\SQLServerReportingServices* is the default SQL Server Reporting Services service account. It is a second-best option, however: we recommend that you create a new domain service account to be used only for this service; for example, *Domain\svc_ServerName_SSRS* or a similar naming convention. You will need to use a domain account if you choose to configure report server email with *Report server service account (NTLM)* authentication.

If you choose to change the SQL Server Reporting Services service account later, use only the Reporting Services Configuration Manager tool to make this change. Like other SQL Server services, never use the Services console (services.msc) to change service accounts.

After installation, you will need to follow-up on other changes and necessary administrative actions; for example, configuring the SQL Server Reporting Services Execution Account, email settings, or backing up the encryption key using Reporting Services Configuration Manager.

SQL Server 2019 Reporting Services also can integrate with Microsoft Power BI dashboards. A page in the Report Server Configuration Manager supports registering this installation of SQL Server Reporting Services with a Power BI account. You will be prompted to sign into Azure Active Directory. The account you provide must be a member of the Azure tenant where you intend to integrate with Power BI. The account should also be a member of the system administrator in SQL Server Reporting Services, via Report Manager, and a member of the sysadmin role in the SQL Server that hosts the Report Server database.

Inside OUT

Where is SQL Server Reporting Services SharePoint Integrated mode?

Starting with SQL Server 2017, there is no more SharePoint Integrated mode. The simplified “native” mode is the only installation available. This matches the moves that Microsoft has made in other areas that step away from the SharePoint on-premises product in favor of SharePoint Online features and development.

Instead, you can integrate SQL Server Reporting Services native mode with on-premises SharePoint sites via embedded SQL Server Reporting Services reports, including SQL Server Reporting Services reports stored in the Power BI Report Server.

Similarly, there is no future support for SQL Server Reporting Services integration with SharePoint Online.

Installing machine learning features

The Machine Learning Services (In-Database or the standalone Machine Learning Server) feature makes it possible for developers to integrate with the R language and/or Python language extensions using standard T-SQL statements.

Data scientists can take advantage of this feature to build advanced analytics, data forecasting, and algorithms for machine learning.

In SQL Server 2019, the Instance Feature Machine Learning Services (In-Database) is renamed to Machine Learning Services and Language Extensions, while the name for the standalone Shared Feature Machine Learning Server (Standalone) is the same.

Machine Learning Services and Language Extensions support Java code execution inside the Database Engine. By default it installs the Zulu Open JRE if you select the Java checkbox

under “Machine Learning Services and Language Extensions” in the feature selection page. If you must use a different Java JRE because of an application dependency, such as the official Oracle JRE, you should add it via a home variable and the `icacls` Windows command, as detailed here: <https://docs.microsoft.com/sql/language-extensions/install/install-sql-server-language-extensions-on-windows>.

- For more on Installing and Leveraging Machine Learning Services, see Chapter 20, “Leveraging Big Data and Machine Learning.”

Installing PolyBase Query Service for External Data

The PolyBase connector is a much-marketed feature for allowing native connectors for external data sources, even non-Microsoft or non-relational database platforms like Oracle, Teradata, MongoDB, Cloudera, or Apache Hadoop.

Using PolyBase EXTERNAL tables, we can use SQL data types and T-SQL queries to seamlessly query data sources in-place, in what Microsoft calls “data virtualization.” This eliminates the need for complex heterogeneous data movement, and reduces the need for developers to have knowledge of other external query languages. The PolyBase Query Engine feature is specifically designed for read and write queries against non-Microsoft database platforms like Oracle and DB2, but also for Hadoop nonrelational data or Azure Blob Storage files, MongoDB, and more. This is a superior alternative to linked servers to the same external data sources, because PolyBase allows “Push Down” computation for these external sources, reducing the amount of data transferred and increasing the performance of analytical-scale queries.

New to SQL Server 2019 is the ability to install the *Java connector for HDFS data sources* with SQL Server Setup. This 7GB+ option doesn’t replace the requirement to install the Oracle JRE in order to install the PolyBase Query Service feature.

- For more on the PolyBase Query Service feature, see Chapter 20.

Logging SQL Server Setup

SQL Server Setup generates a large number of logging files for diagnostic and troubleshooting purposes. These logs should be the first place you go when you have an issue with Setup.

First, a System Configuration Check report .htm file is generated each time you run Setup, so you can view this report in SQL Server Setup near the start of the installation steps.

A new timestamp-named folder of log files is generated for each launch of SQL Server Setup. After you proceed past the *Ready To Install* page, and regardless of whether Setup was a complete success, it generates a number of log files in the following folder:

```
%programfiles%\Microsoft SQL Server\150\Setup Bootstrap\Log\YYYYMMDD_HHMMSS\
```

However, when you run Setup using the /Q or /QS parameters for unattended installation, the log file is written to the Windows %temp% folder.

A log summary file of the installation is created that uses the following naming convention:

Summary_*instancename*_YYYYMMDD_HHMMSS.txt

Setup generates similar files for the Component and Global Rules portions of Setup as well as a file called Detail.txt in the same folder. These files might contain the detailed error messages you are looking for when troubleshooting a failed installation. The Windows Application Event log might also contain helpful information in that situation.

You'll also find the new SQL Server instance's first error log encoded at UTC time in this folder, showing the log from startup, similar to the normal SQL Server Error Log.

NOTE

At the time of this writing, there is an outstanding installation issue with SQL Server 2019, installation media mounted on removeable media, and SQL Server Management Studio versions 18.0 through 18.3. The problem and solution are detailed here in a link provided by Microsoft: <https://techcommunity.microsoft.com/t5/SQL-Server-Support/SQL-Server-2019-Installation-Error-An-error-occurred-for-a/ba-p/998033>. In short: copy the files locally, or, uninstall the SQL Server 2012 Native Client, then re-try installation of SQL Server 2019. You should always try to run the latest version of SSMS.

Automating SQL Server Setup by using configuration files

Let's dig more into what you can do with setup.exe outside of the user interface. You can use configuration files to automate the selection process when installing SQL Server, which helps to create a consistent configuration.

Values provided in configuration files can prepopulate or override Setup settings. They also can configure Setup to run with the normal user interface, or silently without any interface.

Starting SQL Server Setup from the command line

You can start setup.exe from either Windows PowerShell or the command prompt, providing repeatability and standardization of parameter options. You also can use it to prefill sections of the Setup wizard or to change the default behavior of Setup.

For the purposes of the installer, ensure that you always use the Administrator level for these two shells. The title on each application window should be preceded by *Administrator:*, such as: *Administrator: Windows PowerShell*.

Sometimes, you also might find it necessary to start Setup from the command line or Windows PowerShell because of a workaround for a specific problem, or to automate and standardize

future SQL Server installations. To start Windows PowerShell or command prompt as Administrator, in the Start menu, search for the desired application (see Figure 4-4), right-click it, and then, on the shortcut menu that opens, select **Run As Administrator**.

```
PS D:\> .\Setup.exe /ConfigurationFile=c:\install\SQL2019_basic.INI
Microsoft (R) SQL Server 2019 15.00.2000.05
Copyright (c) 2019 Microsoft. All rights reserved.
```

Figure 4-4 Starting Setup.exe from PowerShell.

From the location of the SQL Server Setup installation files, for example the mounted .iso file, execute the following with PowerShell or the Windows Prompt:

```
.\Setup.exe /ConfigurationFile=c:\install\SQL2019_basic.INI
```

The previous sample script and all scripts for this book are available for download at <https://www.microsoftpressstore.com/SQLServer2019InsideOut/downloads>. The above code sample uses a configuration file to pre-select installation choices, for example, features to be installed. Let's talk more about configuration files.

Generating a configuration file

Writing a configuration file by hand is not necessary and can be tedious. Instead of going through that effort, you can let SQL Server Setup create a configuration file for you.

Work your way through the normal SQL Server Setup user interface, completing everything as you normally would, but pause when you get to the *Ready To Install* page. Near the bottom of this page is a path (see Figure 4-5). At that location, even before you hit the **Install** button, you'll find a generated configuration file, ready for future use and modification if needed.

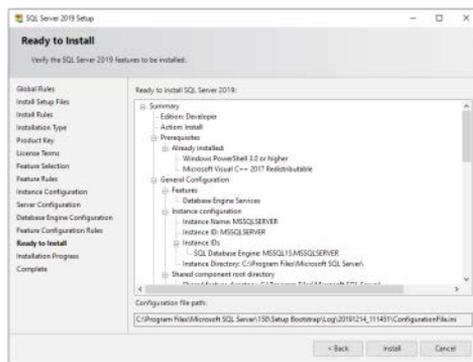


Figure 4-5 The Ready To Install page displays a summary of the installation steps as the Configuration File Path that has been prepared.

For example, the first modification you need to make to the .ini file is to accept the SQL Server license terms via the IACCEPTSSQLSERVERLICENSETERMS parameter, which isn't automatically

provided in the automatically generated .ini file. Unless you modify an .ini file to provide this, it isn't possible to run the installer without user interaction.

Installing by using a configuration file

Now that you have a configuration file generated by using a previous walk-through of Setup, you can take the next step to automating or standardizing your installation.

You can start Setup.exe from a Prompt with a configuration file by using the /CONFIGURATIONFILE parameter of Setup.exe. Or, you can launch SQL Server Setup with a configuration file by navigating to the Advanced page of the SQL Server Installation Center that starts with Setup.exe in Windows. Select the *Install Based On A Configuration File* check box. A message appears, asking you to browse to the .ini file. After you select the appropriate file, setup.exe will start with those options.

One thing to keep mind, however, is that configuration files generated by Setup.exe do not and should not store the passwords you provided for any service accounts. If you do want to configure service account credentials in your configuration file, for security reasons, do not store the service account passwords in plain text in a configuration file. You should instead store passwords separately and securely and provide them when you run setup.exe.

Each service's account parameters are available in a Setup.exe runtime parameter, which is listed in Table 4-1.

Table 4-1 Common Setup.exe parameters and their purposes

Service	Parameter name	Description
SQL Server Database Engine	/SQLSVCPASSWORD	Password for the SQL Server Database Engine Services service account. This is the service account for sqlservr.exe. It is required if a domain account is used for the service.
SQL Server Agent	/AGTSVCPASSWORD	Password for the SQL Server Agent service account. This is the service account for sqlagent.exe. It is required if a domain account is used for the service.
sa password	/SAPWD	Password for the sa account. It is required when /SECURITYMODE=SQL is used, which enables Mixed Mode authentication.
Integration Services	/ISSVCPASSWORD	Password for the Integration Services service. It is required if a domain account is used for the service.
Reporting Services (Native)	/RSSVCPASSWORD	Password for the Reporting Services service. It is required if a domain account is used for the service.

Analysis Services	/ASSVCPASSWORD	Password for the Analysis Services service account. It is required if a domain account is used for the service.
PolyBase	/PBDMSSVCPASSWORD	Password for the PolyBase engine service account.
Full-Text filter launcher service	/FTSVCPASSWORD	Password for the Full-Text filter launcher service.

For example, in the snippet that follows, the `PROD_ConfigurationFile_Install.INI` has provided the account name of the of the SQL Server Database Engine service account, but the password is provided when `Setup.exe` runs in the command prompt or PowerShell:

```
Setup.exe /SQLSVCPASSWORD="securepwd" /ConfigurationFile="d:\SQL\PROD_Install.INI"
```

The previous sample script and all scripts for this book are available for download at <https://www.MicrosoftPressStore.com/SQLServer2019InsideOut/downloads>.

You can provide further parameters like passwords when you run `Setup`. Parameter settings provided will override any settings in the configuration file, just as the configuration file's settings will override any defaults in the `Setup` operation. Table 4-2 lists and describes the parameters.

By default the `/UpdateEnabled` parameter is enabled and doesn't need to be specified, and SQL Server will include updates found via Windows Update. If you choose to disable this behavior by provided `/UpdateEnabled=False`, you can also specify `/UpdateSource` the location of cumulative update or other SQL patch file executables to be included in the installation.

Table 4-2 Common `Setup.exe` parameters of which you should be aware

Parameter usage	Parameter	Description
Unattended installations	/Q	Specifies Quiet Mode with no user interface and user interactivity allowed.
Unattended installations	/QS	Specifies Quiet Mode with user interface but no user interactivity allowed. Will fail if all needed information or parameters are not provided.
Accept license terms	/IACCEPTSQLSERVERLICENSETERMS	Must provide in any Configuration File looking to avoid prompts for installation.
R open license terms	/IACCEPTROPENLICENSETERMS	Must provide this parameter for any unattended installation involving the R language option for Machine Learning Services.

Python open license terms	/IACCEPTPYTHONLICENSETERMS	Must provide this parameter for any unattended installation involving the Python language option for Machine Learning Services.
Instant file initialization	/SQLSVCINSTANTFILEINIT	Set to true to Grant Perform Volume Maintenance Task privilege to the SQL Server Database Engine Service (recommended).
Windows accounts to provision as members of the sysadmin role	/SQLSYSADMINACCOUNTS	Must provide groups or service accounts to specify as the initial members of the sysadmin role.
Provision the user running SQL Server Setup as a member of the sysadmin role	/ADDCURRENTUSERASSQLADMIN	If desired, also specify the current local Windows Server user running SQL Server Setup as an initial member of the sysadmin role. Not desired if using a personal named account, use a group instead.
TempDB data file count	/SQLTEMPDBFILECOUNT	Set to the number of desired TempDB data files to be installed initially.
Enable the TCP/IP Protocol	/TCPENABLED="1"	Disabled by default and used in many installations, enable the TCP/IP protocol here to save yourself a step in Configuration Manager later on.

Post-installation server configuration

After you install SQL Server, there are a number of changes to make or confirm on the Windows Server and in settings for SQL Server.

Post-installation checklist

You should run through the following checklist on your new SQL Server instance. The order of these items isn't necessarily specific, and many deal with SQL server and/or Windows configuration settings. Evaluate whether these are appropriate for your environment, but you should consider and apply them to most SQL Server installations.

1. Check your SQL Server patch level version, and apply patches if necessary.
2. Review Maximum server memory settings for other features.
3. Review Surface Area Configuration.
4. Set up SQL Agent.

5. Turn on TCP/IP if needed.
6. Verify power options for the server.
7. Configure antivirus exclusions for SQL Server processes and files.
8. Enable the server setting “optimize for ad hoc workloads.”
9. Evaluate whether Lock pages in memory is necessary.
10. Review size and location of the Windows page file.
11. Set up scheduled backups, index maintenance, log retention maintenance, and integrity checks.
12. Backup service master and database master keys.
13. Increase SQL Agent and SQL Error log retention from the defaults.
14. Suppress successful backup messages.

Let’s take a look at each of these in more detail in the subsections that follow.

Check your SQL Server version

After you install SQL Server, check the version number against the latest cumulative updates list, especially if you did not opt to or could not use Windows Update during SQL Server Setup. You can view the version number in SQL Server Management Studio’s Object Explorer or via a T-SQL query on either the following built-in functions:

```
SELECT @@VERSION;  
SELECT SERVERPROPERTY('ProductVersion');  
SELECT SERVERPROPERTY('Edition');
```

While you’re at it, doublecheck that you installed the right edition of SQL Server too!

NOTE

Take the opportunity before your SQL Server enters production to patch it. For information about the latest cumulative updates for SQL Server, search for KB321185 or visit <https://support.microsoft.com/help/321185/>.

Maximum server memory settings for other features

Other features of SQL Server have their own maximum server memory settings. As you will notice by their default settings, for servers on which both the SQL Server Database Engine and SQL Server Analysis Services and/or SQL Server Reporting Services are installed, competition for and exhaustion of memory is possible. It is recommended that you protect the Database Engine by lowering the potential memory impact of other applications.

Limiting SQL Server Analysis Services memory

SQL Server Analysis Services (SSAS) has not just one maximum server memory limit, but five, and you can enforce limits by hard values in bytes or by a percentage of total physical memory of the server.

You can change these memory settings via SSMS by connecting to the SSAS instance in Object Explorer. Right-click the server, and then, on the shortcut menu, click **Properties**. Some of the memory settings described here are identical for Multidimensional and Tabular installations, some are for Tabular mode only:

- **LowMemoryLimit.** A value that serves as a floor for memory, but also the level at which SSAS begins to release memory for infrequently used or low-priority objects in its cache. Below this level no memory maintenance is performed by SSAS. The default value is 65, or 65 percent of total server physical memory (or the Virtual Address Space technically, but SSAS, among other features, is no longer supported on 32-bit systems, and so this is not a concern).
- **TotalMemoryLimit.** A value that serves as a threshold for SSAS to begin to release memory for higher priority requests. It's important to note that this is not a hard limit. The default is 80 percent of total server memory.
- **HardMemoryLimit.** This is a hard memory limit that will lead to more aggressive pruning of memory from cache and potentially to the rejection of new requests. By default, this is displayed as 0 and is effectively the midway point between the `TotalMemoryLimit` and the server physical memory. The `TotalMemoryLimit` must always be less than the `HardMemoryLimit`.
- **VertiPaqMemoryLimit.** For SSAS installations in Tabular mode only, this limit has a default of 60, or 60 percent of server physical memory. Above this percentage, and only if `VertiPaqPagingPolicy` is turned on (it is by default), SSAS begins to page data to the hard drive using the OS page file. Paging to a drive can help prevent out-of-memory errors when the `HardMemoryLimit` is met.
- **QueryMemoryLimit.** New to SQL Server 2019, this can limit the amount of memory used by individual DAX queries, preventing any one query from dominating memory. For any individual query, this setting can be overridden by a new XMLA property `DbpropMsmdRequestMemoryLimit`, specified for the query connection. This setting can be specified as a percentage (values ≤ 100), or as a number of bytes greater than 100. The default setting of 0 implies no limit to the memory of individual queries.

Figure 4-6 shows the General page of the Analysis Server Properties dialog box, as started in Object Explorer in SSMS, and the locations of the preceding memory configuration properties with their defaults in SQL Server 2019 for a Tabular mode installation of SSAS. Note that the option to “Show Advanced (All) Properties” is checked.

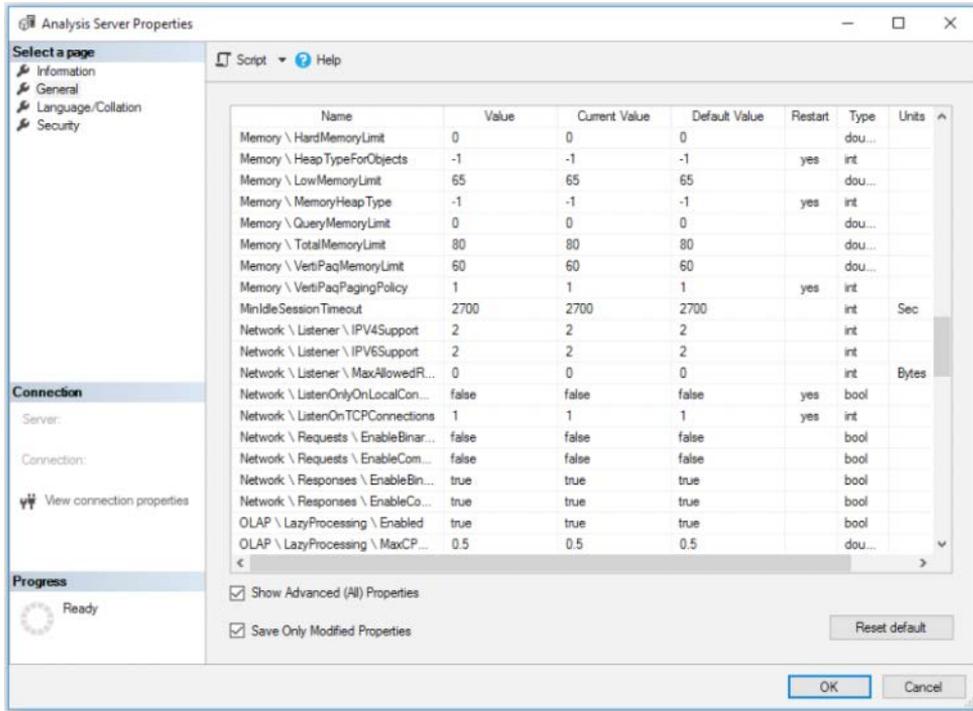


Figure 4-6 The General page in the Analysis Server Properties dialog box showing the default settings.

Limiting SQL Server Reporting Services memory

Four options are available for limiting SQL Server Reporting Services memory utilization, and all are based on numbers contained in tags within a .config file, so be sure to make a backup of it before editing. You can configure memory settings only in the *RSReportServer.config* file, which is a text file that is stored at *%ProgramFiles%\Microsoft SQL Server Reporting Services\SSRS\ReportServer*.

NOTE

This location has changed from previous versions, but the config file name has not.

Two of the settings are in the .config file by default; two more are available to administrators to use in advanced scenarios.

Let's take a look at each one:

- MemorySafetyMargin.** The percentage of *WorkingSetMaximum* that SQL Server Reporting Services will use before taking steps to reduce background task memory

utilization and prioritize requests coming from the web service, attempting to protect user requests. User requests could still be denied.

- **MemoryThreshold.** The percentage of `WorkingSetMaximum` at which SQL Server Reporting Services will deny new requests, slow down existing requests, and page memory to a hard drive until memory conditions improve.

Two more settings are instead given values automatically upon service startup, but you can override them in the `.config` file. Two older memory settings from SQL Server 2005 with which SQL DBAs might be familiar, are `MemoryLimit` and `MaximumMemoryLimit`, but those two values have been ignored since SQL Server 2008.

- **WorkingSetMaximum.** By default, this is the total server's physical memory. This setting does not appear by default in the `.config` file, but you can override it to reduce the amount of memory of which SQL Server Reporting Services will be aware. This value is expressed in kilobytes of memory.
- **WorkingSetMinimum.** By default, this value is 60 percent of the `WorkingSetMaximum`. If SQL Server Reporting Services needs memory below this value, it will use memory and not release it due to memory pressure. This setting does not appear by default in the `.config` file, but you can override it to increase the variability of SQL Server Reporting Service's memory utilization.

These four settings can appear in the `rsreportserver.config` file. As demonstrated here, you could override the default settings to 4 GB maximum and 2 GB minimum (each expressed in KB):

```
<MemorySafetyMargin>80</MemorySafetyMargin>
<MemoryThreshold>90</MemoryThreshold>
<WorkingSetMaximum>4194304</WorkingSetMaximum>
<WorkingSetMinimum>2097152</WorkingSetMinimum>
```

Limiting Machine Learning Server memory

Similar to SSAS and SSRS, the Machine Learning Server has a `.config` file at `%ProgramFiles%\Microsoft SQL Server\MSSQL15.instanceName\MSSQL\Binn\rlauncher.config`.

By default, Machine Learning Server is similar to 20 percent of total server memory. You can override this by adding a tag to the `.config` file to provide a value for `MEMORY_LIMIT_PERCENT`. This value is not in the `.config` file by default.

Remember to make a backup of this config file before editing. The following is an example of the contents of the `rlauncher.config` file, with the default memory limit changed to 25 percent:

```
RHOME=C:\PROGRA~2\MICROS~1\MSSQL1~4.SQL\R_SERV~2
MPI_HOME=C:\Program Files\Microsoft MPI
INSTANCE_NAME=SQL2K19
```

```
TRACE_LEVEL=1
JOB_CLEANUP_ON_EXIT=1
USER_POOL_SIZE=0
WORKING_DIRECTORY=C:\Program Files\Microsoft SQL
Server\MSSQL15.SQL2K19\MSSQL\ExtensibilityData
PKG_MGMT_MODE=0
MEMORY_LIMIT_PERCENT=25
```

The previous sample script and all scripts for this book are available for download at <https://www.MicrosoftPressStore.com/SQLServer2019InsideOut/downloads>.

Surface Area Configuration

If you are a veteran SQL Server DBA, you will remember when SQL Server Surface Area Configuration was a separate application. Surface Area Configuration is now considered a facet, accessed via the Facets dialog in SSMS starting with SQL Server 2008.

To view Surface Area Configuration in SSMS, in Object Explorer, connect to the *SQL Server*, right-click the server, and then, on the shortcut menu, click **Facets**. (Note that this window sometimes takes a moment to load.) In the dialog box that opens, change the value in the list box to **Surface Area Configuration**.

Keep in mind that most of these options should remain off unless needed because they present a specific potential for misuse by an administrator or unauthorized user. In typical installations of SQL Server 2019, however, you should consider enabling three of these options:

- Database Mail (more about this in Chapter 9: "Automating SQL Server administration"). This should be enabled on most instances, to allow SQL Server at the very least to send out a message in case of a high severity or job failure, and also to allow developers to send custom email message using the system procedure `sp_send_dbmail`. You also can turn this setting on or off via the Database Mail XPs option in `sp_configure`.
- Remote Dedicated Admin Connection (more on this in Chapter 13: "Protecting data through encryption, privacy and auditing"). This could be particularly useful for bypassing a malfunctioning login triggers or Resource Governor. You also can turn this setting on or off via the remote admin connections option in `sp_configure`.
- CLR Integration, which you will need to turn on if you need to use SQL Server Integration Services, or if you need to write CLR objects. You also can turn this setting on or off via the `clr_enabled` option in `sp_configure`.

You should turn on other options in Surface Area Configuration only if they are specifically required by an application and you are aware of the potential security concerns.

Setting up SQL Agent

There are several post-installation tasks to set up in SQL Agent before SQL Server can begin to help you automate, monitor, and back up your new instance.

► **Chapters 8 and 9 cover SQL Agent and monitoring topics in detail.**

You will likely want to do the following:

1. Change the SQL Agent service from Manual to Automatic startup.
2. To send email notifications for alerts or job status notifications, you must set up a Database Mail account and profile (see Chapter 9).
3. Set up an Operator for a distribution group of IT professionals in your organization who would respond to a SQL Server issue.
4. Configure SQL Server Agent to use Database Mail, including a fail-safe operator.
5. Set up SQL Server Alerts for desired errors and high severity (Severity 21+) errors.

At the very least, these steps are put in place so that SQL Server can send out a call for help. Even if you have centralized monitoring solutions in place, the most rare and severe of errors should be important enough to warrant an email.

You can choose to configure a large number of Windows Management Instrumentation (WMI) conditions, Perfmon counter conditions, and SQL Server Error messages by number or severity in SQL Server Alerts. However, do not overcommit your inboxes, and do not set an inbox rule to Mark As Read and file away emails from SQL Server. By careful selection of emails, you can assure yourself and your team that emails from SQL Server will be actionable concerns that rarely arrive.

► **For much more information on Managing and Monitoring SQL Server, see Chapter 13.**

Enabling the TCP/IP network protocol

The common network protocol TCP/IP is off by default, and the only protocol that is on is Shared Memory, which allows only local connections. You will likely not end up using Shared Memory alone to connect to the SQL Server for common business applications that use multiple servers for database, web, and application tiers.

NOTE

It is possible to enable the TCP/IP protocol by default at the time of installation if using a configuration file for SQL Server Setup, but this option does not appear in the UI for SQL Server setup. It must be changed after installation is complete.

When you connect to SQL Server using SSMS while local to the server, you connect to the Shared_Memory endpoint whenever you provide the name of the server, the server\instance, localhost, the dot character ("."), or (local), or .\instance, or (local)\instance.

TCP/IP, however, is ubiquitous in many SQL Server features and functionality. Many applications will need to use TCP/IP to connect to the SQL Server remotely. Many SQL Server features require TCP/IP to be enabled, including the Remote Dedicated Admin Connection (DAC), the Always On availability groups listener, and Kerberos authentication.

To configure the TCP/IP protocol, open the SQL Server Configuration Manager application locally on the server, in the left pane, click **SQL Server Network Configuration**. Browse to the protocols for your newly installed instance of SQL Server. The default instance of SQL Server, here and in many places, will appear as **MSSQLSERVER**.

You can also enable the TCP/IP protocol for a SQL Server instance with PowerShell:

```
Import-Module SqlServer
$wmi = new-object('Microsoft.SqlServer.Management.Smo.Wmi.ManagedComputer')
#Path to the local server
$path = "ManagedComputer[@Name='$env:COMPUTERNAME']/"
$path = $path+"ServerInstance[@Name='SQL2K19']/ServerProtocol[@Name='Tcp']"
#Enable the TCP protocol on the local server, on the named instance SQL2K19
$TCPIP = $wmi.GetSmoObject($path)
$TCPIP.IsEnabled = $true
$TCPIP.Alter()
$TCPIP.IsEnabled
#Restart SQL Server Database Engine service to apply the change
```

The previous sample script and all scripts for this book are available for download at <https://www.MicrosoftPressStore.com/SQLServer2019InsideOut/downloads>.

After turning on the TCP/IP protocol via any method, you need to restart the SQL Server Database Engine service for it to take effect.

NOTE

Turning on Named Pipes is not required or used unless an application specifically needs it.

Verifying server power options

The Windows Server Power Options setting should be set to High Performance for any server hosting a SQL Server instance.

In other power plans, Windows might not operate the processor at maximum frequency during normal or even busy periods of SQL Server activity. This applies to physical or virtual Windows servers.

Review this setting and ensure that the group policy will not change this setting back to Balanced or another setting. Also ensure that group preferences are configured for High Performance for new SQL Servers.

Configuring antivirus exclusions

Configure any antivirus software installed on the SQL Server to ignore scanning files with extensions used by your SQL Server data and log files. Typically, these will be .mdf, .ldf, and .ndf.

Also, configure any antivirus programs to ignore folders containing SQL Server files. This could include:

- Full-text catalog files
- Backup files
- Replication snapshot files
- SQL Server trace (.trc) files
- SQL Audit files
- Analysis Services database
- Log and backup files
- FILESTREAM and FileTable folders
- SQL Server Reporting Services temp files and log files

Processes might also be affected, so set antivirus programs to ignore the programs for all instances of the SQL Server Database Engine service, Reporting Services service, Analysis Services service, and R Server (RTerm.exe and BxlServer.exe).

In SQL Server Failover Cluster Instances (FCIs) and availability groups, also configure antivirus software to exclude the MSCS folder on the quorum drive if in use, the MSDTC directory on the MSDTC share, and the Windows\Cluster folder on each cluster node, if they exist.

Inside OUT

What if you suspect antivirus or antimalware software is interfering with SQL Server?

This is one of the more challenging troubleshooting exercises: a strange error message, DLL error, or file accessibility issue. It is critical to configure antivirus to exclude SQL Server files and folders from on-access scans, exclusive-lock scans, and more.

If you notice, for example, random databases failing to recover upon SQL Server startup, or error messages regarding *File activation failure* or *Unable to open the physical file*, sqlservr.exe may not be able to gain exclusive access to the files because they are being scanned by another application. Use the Windows sysinternals Process Explorer application to search for handles including your SQL Server files, and potentially catch that other application accessing the file.

Antivirus applications may also interfere with service packs and cumulative updates if those files, even if they are signed by Microsoft, have not been pre-approved for execution in the production environment. Communicate with the teams that control antivirus, anti-ransomware, or anti-malware solutions in your enterprise.

Enable Optimize For Ad Hoc Workloads

The server-level setting *Optimize For Ad Hoc Workloads* doesn't have the most intuitive name.

We are not optimizing ad hoc queries; we are optimizing SQL Server memory usage to prevent ad hoc queries from consuming unnecessary cache.

- ▶ For more about the *Optimize For Ad Hoc Workloads* setting, see Chapter 2: Introducing database server components.

For the unlikely scenario in which a large number of queries are executed only two times, setting this option to **True** would be a net negative for performance. However, like other design concepts in databases, we find that there are either one or many. There is no two.

- ▶ To read more about cached execution plans, see Chapter 14.

Lock Pages in Memory

You should consider using the *Lock Pages in Memory* setting for environments in which instances of SQL Server are expected to experience memory pressure due to other applications, server limitations, or overallocated virtualized systems. This is an in-depth topic to be carefully considered.

- ▶ For more about the *Lock Pages in Memory* setting, see Chapter 2.
- ▶ For more about the Windows page file, see Chapter 3.

Inside OUT

How can you tell if the permission to Lock Pages in Memory is in effect?

Starting with SQL Server 2016 SP1, you can check whether the *Lock Pages in Memory* permission has been granted to the SQL Server Database Engine service:

```
SELECT sql_memory_model_desc
FROM sys.dm_os_sys_info;
--CONVENTIONAL = Lock pages in memory privilege is not granted
--LOCK_PAGES = Lock pages in memory privilege is granted
--LARGE_PAGES = Lock pages in memory privilege is granted in Enterprise mode
-- with Trace Flag 834 ON
```

The previous sample script and all scripts for this book are available for download at <https://www.MicrosoftPressStore.com/SQLServer2019InsideOut/downloads>.

Backups, index maintenance, and integrity checks

Backups are a critical part of your disaster recovery, and they should begin as soon as possible after installation, and before the users or applications begin to use the SQL Server.

Begin taking database backups, at least of the master and msdb databases. You should also back up other SQL Server Setup-created databases, including ReportServer, ReportServer-TempDB, and SSISDB right away.

- For more information on backups, index maintenance, and monitoring, see Chapter 11.

As soon as your new SQL Server instance has databases in use, regularly perform selective index maintenance and integrity checks that take into account the current fragmentation levels of indexes, rather than performing index maintenance on entire databases.

- For more information on automating maintenance, see Chapter 13.

Backing up service master and database master keys

You should back up service master keys and any database master keys as they are created, securely storing their information.

- For more information on service master and database master keys, see Chapter 6.

To back up the instance service master key, use the following:

```
BACKUP SERVICE MASTER KEY TO FILE = 'localfilepath_or UNC' ENCRYPTION BY PASSWORD = 'complexpassword'
```

And as soon as database master keys come into existence in each user database, for example as you implement features like Transparent Data Encryption (TDE) or column data encryption, back up individual database master keys, as follows:

```
BACKUP MASTER KEY TO FILE = 'localfilepath_or UNC' ENCRYPTION BY PASSWORD =
'complexpassword'
```

The previous sample script and all scripts for this book are available for download at <https://www.MicrosoftPressStore.com/SQLServer2019InsideOut/downloads>.

If you implement Transparent Data Encryption, Always Encrypted, native backup encryption, column encryption, or any other native or external solutions that generate certificates, keys and/or passwords, develop a secure storage and retrieval method inside your enterprise. Failure to back up master and database master keys could compromise future disaster recovery attempts!

Increasing default error and agent history retention

By default, SQL Server maintains the current SQL Server error log plus six more error logs of history. Logs are cycled each time the SQL Server service is started.

One eventful, fun weekend of server troubleshooting or maintenance where the SQL Server service is restarted many times could wipe out a significant amount of your error history. This could make the task of troubleshooting periodic or business-cycle related errors difficult or impossible. You need visibility into errors that occur only during a monthly processing, monthly patch day, or periodic reporting.

In SQL Server Management Studio, in Object Explorer, connect to the SQL Server instance. Expand the Management folder, right-click **SQL Server Logs**, and then, on the shortcut menu, click **Configure**. Select the *Limit The Number Of Error Logs Before They Are Recycled* check box and type a value larger than 6. You might find that a value between 25 and 50 will result in more useful log history contained for multiple business cycles.

On SQL Servers that generate a large amount of log “noise”, consider other options to reduce the clutter of the SQL Error Log, including Trace Flag 3226 to suppress the logging of successful backup operations (much more on that in the next section).

You may also choose to configure a SQL Agent Job to manually cycle the SQL Error Log using `sp_cycle_errorlog`, so that no one log file contains too much data to be unwieldy for scan and analysis.

Similarly, you might find that the SQL Server Agent history is not sufficient to cover an adequate amount of job history, especially if you have frequent job runs.

To change the history settings for SQL Server Agent, in SSMS, in Object Explorer, connect to the SQL Server instance. Right-click **SQL Server Agent**, and then click **Properties**. Click the

History page. This page is not intuitive and can be confusing. The first option, *Limit Size Of The Job History Log*, is a rolling job history retention setting. Consider increasing the maximum log history size in rows from the default of 1,000 to 10,000 or more, and also increase the maximum job history per job in rows from the default of 100 to 2,000 or more. This data is stored in the msdb system database and will cause that database to grow larger over time. Consider pre-allocating now some additional file space to the msdb data file.

The second option, *Remove Agent History*, along with its companion *Older Than* text box is not a rolling job history retention setting; rather, it is an immediate and manual job history pruning. Select this second check box, and then click **OK** and return to this page; you will find the second check box is cleared. Behind the scenes, SQL Server Management Studio ran the `msdb.dbo.sp_purge_jobhistory` stored procedure to remove job history manually.

Suppress successful backup messages

By default, SQL Server writes an event to the error log upon a successful database backup, whether it be FULL, DIFFERENTIAL, or TRANSACTION LOG.

On instances with many databases and with many databases in FULL recovery model with regular transaction log backups, the amount of log activity generated by just their successful frequent log backups could flood the log with clutter, lowering log history retention.

NOTE

It is important to note that you can review successful backup history by querying the msdb system database, which has a series of tables dedicated to storing the backup history for all databases, including `msdb.dbo.backupset` and `msdb.dbo.backupmedialy`. The built-in “Backup and Restore Events” report in SQL Server Management Studio provides access to this data, as well.

- For more on backups, see Chapter 10, “Developing, deploying, and managing data recovery.”

SQL Server Trace Flag 3226 controls an option at the instance level to suppress successful backup notifications.

There are many trace flags available to administrators to alter default behavior—many more options than there are user interfaces to accommodate them in SQL Server Management Studio. Take care when turning them on and understand that many trace flags are intended only for temporary use when aiding troubleshooting.

As Trace Flag 3226 is intended to be a permanent setting, simply starting the trace by using `DBCC TRACEON` is not sufficient, given that the trace flag will no longer be active following a SQL Server service restart. Instead, add the trace flag as a startup parameter to the SQL Server Database Engine service by using SQL Server Configuration Manager. In the Properties of the SQL

Server service, go to the Startup Parameters tab, and use the syntax `-Tflagnumber`. This field is essentially adding parameters that are passed to the `sqlserver.exe` executable. For example, add `"-T3226"`, then click Add. The change will not take effect until the SQL Server Database Engine service is restarted.

► For more information on SQL Server Configuration Manager, see Chapter 1.

Post-installation configuration of other features

SQL Server Database Engine installation is now complete, but three other features require post-installation configuration, including SQL Server Integration Services, SQL Server Reporting Services, and SQL Server Analysis Services. You will need to perform the steps detailed in this section before use if these features are installed.

SSISDB initial configuration and setup

Among the best features added by SQL Server 2012 were massive improvements to SQL Server Integration Services, specifically with a new server-integrated deployment, built-in performance data collector, environment variables, and more developer quality-of-life improvements. For these reasons, use the new Project Deployment Model and the built-in SSISDB for all new development.

When the Integration Services Catalog is created, a new user database called SSISDB is also created. You should back it up and treat it as an important production database.

You should create the SSISDB catalog soon after installation and before a SQL Server Integration Services development can take place. You will need to create the catalog only once. Because this will involve potential Surface Area Configuration changes and the creation of a new strong encryption password, a SQL DBA, not a SQL Server Integration Services developer, should perform this and store the password securely alongside others generated at the time of installation.

In Object Explorer, connect to your instance, right-click **Integration Services Catalog**, and then, on the shortcut menu, click **Create Catalog**. In this single-page setup, you must select the *Enable CLR Integration* check box, decide whether SQL Server Integration Services packages should be allowed to be run at SQL Server Startup (we recommend this due to the maintenance and cleanup performed then), and provide an encryption password for the SSISDB database.

The encryption password is for the SSISDB database master key. After you create it, you should then back up the SSISDB database master key.

► For more on database master keys, see Chapter 13.

The SSISDB database will contain SQL Server Integration Services packages, their connection strings, and more data about the packages. The encryption would not allow these sensitive

contents to be decrypted by a malicious user who gains access to the database files or backups. This password would be required if the database were moved to another server, so you should store it in a secure location within your enterprise.

NOTE

If you receive an error when creating the SSSIDB catalog that reads “The catalog backup file ‘C:\Program Files\Microsoft SQL Server\150\DTS\Binn\SSISDBBackup.bak’ could not be accessed” or similar, it is likely because SQL Server Integration Services was not actually installed. It’s likely that the 6MB template database backup was not copied from the SQL Server media, perhaps because the SSIS feature was not installed on this instance. You can run SQL Server Setup again or copy the SSISDBBackup.bak file from another SQL Server installation of the same version.

SQL Server Reporting Services initial configuration and setup

There are still tasks to perform upon first installation of a SQL Server Reporting Services native-mode installation from the downloaded installer file, `SQLServerReportingServices.exe`. At the end of the Microsoft SQL Server 2019 Reporting Services installer wizard, on the Setup Completed screen, click the **Configure report server** button to open the Reporting Services Configuration Manager application. Connect to the newly installed SQL Server Reporting Services instance, and then review the following options, from top to bottom:

- **Service Account.** You can change the SQL Server Reporting Services service account here. Remember that you should use only the Reporting Services Configuration Manager tool to make this change.
- **Web Service URL.** The web service URL is not for user interaction; rather, it is for the Report Manager and custom applications to programmatically connect to the SQL Server Reporting Services instance.

By default, a web service on TCP Port 80 is created called `ReportServer`. For named instances, the web service will be called `ReportServer_instancename`. The URL for the webservice would then be:

`servername/ReportServer`

or:

`servername/ReportServer_instancename`

To accept defaults, at the bottom of the application window, click **Apply**.

You can optionally configure an SSL certificate to a specific URL for the Web Portal in the Advanced section here. Choose an identity and an HTTPS certificate that’s been loaded to the server, and the Reporting Services Configuration Manager will make the changes necessary. For more information on configuring SSL connections for the SSRS

Web Service and Web Portal, visit: <https://docs.microsoft.com/sql/reporting-services/security/configure-ssl-connections-on-a-native-mode-report-server>.

- **Database.** Each instance of SSRS requires a pair of databases running on a SQL Server instance. Simply executing the SSRS installer alone does not configure the databases for SSRS, you need to configure them via the Reporting Services Configuration Manager. The database names by default are ReportServer and ReportServerTempDB, or, for a named instance, ReportServer\$*InstanceName* and ReportServer\$*InstanceName*TempDB. Both of these databases are important and you should create a backup schedule for each. The ReportServerTempDB is not a completely transient database like the SQL Server instance's TempDB system database.

The databases for SSRS can be hosted on a on-premises SQL Server instance or Azure VM-hosted SQL Server instance or, new to SQL Server 2019, hosted on an Azure SQL Managed Instance.

To set the databases for a new instance of SSRS, in the Database page of the Reporting Services Configuration Manager, click Change Database, and then follow the *Report Server Database Configuration Wizard*.

- **Web Portal URL.** The web portal URL is the user-friendly website that hosts links to reports and provides for administrative features to the SQL Server Reporting Services instance. This is the link to share with users if you will be using the SQL Server Reporting Services portal.

By default, the URL for the web portal is */Reports*
servername/Reports for the default instance

or:

servername/Reports_InstanceName for named instances

You can change the name from the default here if desired. To proceed, at the bottom of the application window, click **Apply**.

- **Email Settings.** You use these email settings for sending reports to user subscribers via email. SQL Server Reporting Services uses its own Email Settings and does not inherit from the SQL Server instance's Database Mail settings. This setting is optional if you do not intend to send reports to subscribers via email.

SQL Server Reporting Services can authenticate to an SMTP server using anonymous (No Authentication), Basic, or NT LAN Manager (NTLM) authentication, which will use the SQL Server Reporting Services service account to authenticate to the SMTP server.

NOTE Keep in mind that enterprise SMTP servers usually have an allow list of IP addresses, and you will need to add this server's IP to this list to relay email.

- **Execution Account.** You can provide this domain account optionally to be used when reports are configured to run on a schedule, to run without credentials (select the Credentials Are Not Required Option), or to connect to remote servers for external images. The execution account should not be the same as the SQL Server Reporting Services service account.

This account should have minimal read-only access to any data sources that will require it. You also can give it EXECUTE permissions for data sources that use stored procedures, but you should never give it any additional SQL Server permissions or let it be a member of any server roles, including sysadmin.
- **Encryption Keys.** Immediately after installation and after the two SQL Server Reporting Services databases have been created, you should back up this instance's encryption keys. This key is used to encrypt sensitive information such as connection strings in the two databases. If the databases are restored to another server and this key is not available from the source server, credentials in connection strings will not be usable and you will need to provide them again for the reports to run successfully on a new server.

If you can no longer locate the backup of a key, back it up again, or rotate the key by using the *Change* operation on this page to replace the key, and then back it up.

To restore the original key to a new server to which the databases have been moved, use the *Restore* operation on this page.
- **Subscription Settings.** Use this page to specify a credential to reach file shares to which report subscriptions can be written. Reports can be dropped in this file share location in PDF, Microsoft Excel, or other formats for consumption.

Multiple subscriptions can use this file share credential, which can be used on this page in a central location.

This account should be different from the SQL Server Reporting Services execution account to serve its purpose appropriately.
- **Scale-Out Deployment.** Visit this page on multiple SQL Server Reporting Services instances to join them together. By using the same SQL Server Reporting Services databases for multiple SQL Server Reporting Services instances, multiple front ends can provide processing for heavy reporting workloads, including heavy subscription workloads. The server names can optionally be used in a network load balancer such as a Network Load Balancing Cluster, or you can distribute workload to each SSRS instance from different applications.

Upon first installation, the Scale-Out Deployment page will show that the instance is "Joined" to a single server scale-out.

Each scale-out instance of SQL Server Reporting Services must use the same settings on the Database page of the Reporting Services Configuration Manager. Connect to each

instance in the scale-out and visit this page by opening it on each SQL Server Reporting Services instance to view the status, add servers to the scale-out, or remove servers.

For more detail on scale-out deployments of SSRS, visit:

<https://docs.microsoft.com/sql/reporting-services/install-windows/configure-a-native-mode-report-server-scale-out-deployment>.

- **PowerBI Integration.** Use this page to associate the SQL Server Reporting Services instance to a Microsoft Power BI account, specifically to an account in Azure Active Directory. The administrator joining the Power BI instance to the SQL Server Reporting Services instance must be:
 - A member of the Azure AD
 - A member of the system administrator role of the SQL Server Reporting Services instance
 - A sysadmin on the SQL Server instance that hosts the SQL Server Reporting Services databases.
- For the latest information on Power BI/SQL Server Reporting Services integration and the latest Azure authentication features, visit <https://docs.microsoft.com/sql/reporting-services/install-windows/power-bi-report-server-integration-configuration-manager>.

SQL Server Analysis Services initial configuration and setup

No additional steps are required after setup to begin using a new SQL Server Analysis Services instance.

You can initiate manual backups of SQL Server Analysis Services databases in Object Explorer in SQL Server Management Studio as well as restore SQL Server Analysis Services databases. Because of the nature of SQL Server Analysis Services databases, their size, and how they are populated, typically they are not backed up on a schedule, but you can do so by passing an XMLA command via a SQL Server Agent job step: type **SQL Server Analysis Services** .

When installing SQL Server Analysis Services, a security group should have been chosen to grant permissions to SQL Server Analysis Services server administrators, granting a team full access to the server.

If you need to add a different group to the administrator role of the SQL Server Analysis Services instance, open SQL Server Management Studio, and then, in Object Explorer, connect to the *Analysis Services* instance. Right-click the server, and then, on the shortcut menu, click **Properties**. On the *Security* page, you can add additional windows-authenticated accounts or groups to the administrator role.



Index

A

- AAD, 860, 880
- Aardvark, 662
- Aaron Swartz, 49
- abreast, 605
- absurd, 669
- academic, 312, 616
- accelerate, 658
- Accelerated Database, 233, 234, 318, 634
- acceleration, 68
- accent, 224
- Access, 51, 54, 60, 68, 69, 339, 341, 360, 637
- Access Control, 123
- AccessPurchaseOrder, 925
- accessosql, 890
- Account Cloud, 496
- Account Retry Attempts, 383
- Account Retry Delay, 383
- AccountsPersonID, 753, 754
- accrued, 747
- accumulate, 334, 407
- ACID, 657
- ACLs, 123, 239, 917
- acronym, 125, 483, 514
- Active Expensive, 37, 39, 700
- actively, 37, 331, 481, 917
- ActiveX, 389
- Activity Monitor, 36, 37, 38, 39, 40, 41, 391, 700, 701
- Actual Execution, 697, 700, 703, 764
- ActualQueryPlan, 718
- Actual Rebinds, 700
- ActualRows, 699
- ActualRowsRead, 699
- acute, 455
- Add Connection, 417
- Add Database, 508
- AddDays, 428
- Add Login, 30
- AddOption, 725
- Add Replica, 521
- Address Resolution, 613
- Add Server, 30
- ADF, 898
- adhere, 260, 549
- Advanced Cluster, 489, 490
- Advanced Data, xvi, 824, 841, 844, 845, 848, 855
- Advanced Encryption, 448, 625, 626
- Advanced Format, 124
- Advanced Package, 193
- Advanced Policies, 504
- Advanced Services, 147, 164
- adverse, 330, 433, 438, 444, 463, 505, 847
- affiliated, 799
- Agent Extension, 795
- Agent Job, 182, 410, 419
- aggravating, 31
- aging, 250
- agonizing, 435
- Ahead Logging, 97
- Ain, 125
- airbags, 677
- akin, 73
- Alert Definition, 42
- Alert System, 384, 398
- Aliens Land, 610
- All Automatic, 211
- All Database, 211
- All Services, 893
- Alphanumeric, 244
- AMD, 59
- ample, 317, 363
- Analysis Server, 173, 174
- Analytics Solution, 371
- Anatomy, 95
- ancestor, 234, 258
- anchor, 567
- annotations, 133
- anomalies, 648, 846
- Anomalous, 846
- anticipate, 86, 503
- Antimalware, 117
- antipatterns, 573
- antivirus, 152, 172, 179, 180, 458
- Any Counter, 42
- AnyTable, 686
- Apache Hive, 914
- Apache Knox, 911
- Apache Kudu, 911
- Apache Spark, 819, 911, 919
- apiVersion, 133, 134, 135
- apparatus, 401, 574
- AppContainer, 933
- Append Output, 394
- Application event, 642
- Application Insights, 369
- ApplicationName, 19
- Application Programming, 622, 623
- aptitude, 193
- aptly, 206
- arcane, 549, 660
- archaeology, 336
- architectural, 226, 668, 800, 829, 860, 913

arduous, 606
 Area Configuration, 171, 184
 Area Network, 64, 66, 70, 145, 402, 482, 627
 Are Recycled, 182
 Armed, 72
 ASC, 710, 754
 ASE, 883, 884, 890
 ASG, 787
 attachVPNGateway, 869
 auditSettings, 842
 Australia Central, 888
 Authentication modes, 539, 549
 Authentication Users, 569
 authoring, 424
 authoritative, 151
 Avg Disk, 367
 AvgQuantity, 767, 768
 awry, 376
 AWS, 884
 AzResourceGroup, 894, 897
 AzSqlDatabase, 829, 830, 832, 833, 850
 AzSqlDatabaseAudit, 840, 842
 Azure AD, 83, 84, 188, 541, 542, 583, 624, 640, 785, 812
 Azure Analysis, 161, 807
 Azure Application, 653
 Azure Authenticator, 541
 Azure Automation, 43, 150, 806, 811, 832
 Azure Backup, 797
 Azure blob, 800, 845, 852, 863, 872, 924, 925
 Azure cloud, 70, 608, 781, 797, 867
 Azure Command, 783, 916
 Azure Database, 217, 819, 871, 872, 888, 889, 891
 Azure Databricks, 819
 Azure Disks, 133
 Azure File, 132
 Azure Firewall, 787
 Azure Government, 781
 Azure Hybrid, 151, 795, 796, 812, 864
 Azure Integration, 163, 898, 899

Azure Key, 150, 390, 624, 633, 636, 637
 Azure Kubernetes, 131, 133, 916
 Azure Log, 367, 369
 Azure Managed, 132, 888
 Azure Marketplace, 150, 151, 371, 651
 Azure Migrate, 888
 Azure Monitor, 367, 368, 369, 841
 Azure Network, 513
 Azure networking, 652, 787
 Azure Replica, 521
 Azure resource, 368, 369, 810, 835, 836
 Azure Runtime, 899
 Azure Security, 818, 846, 881
 Azure service, 835, 846, 866
 Azure Stack, 815
 Azure subscription, 782, 785, 825
 Azure Support, 827
 Azure Synapse, 25, 44, 541, 542, 788, 797, 798, 818, 912, 924
 Azure Virtual, 7, 649, 650, 651, 652, 788, 789, 794, 871
 Azure web, 811, 816
 Azure Windows, 144

B

backbone, 837, 871
 backdoor, 563
 backend, 513, 925
 backtick, 48
 Backup Devices, 30
 Backup Encryption, 440
 Backup Preferences, 414, 415, 514
 Balancing Cluster, 187
 barely, 438
 Based Access, 785, 824, 838, 839
 based authentication, 542
 Basic Reference, 425
 batch mode, 733, 762, 764
 BatchParser, 429
 Batch Requests, 37, 364
 BCHR, 397
 BCM, 95

BCP, 8, 12, 13, 14, 197, 434, 438
 Benefit Savings, 812
 benign, 334
 BGP, 615, 651
 bigint, 107, 249, 264, 359, 744, 745
 binaries, 143, 202, 345, 489
 binding, 137
 Bing Maps, 254
 BIOS, 61
 bite, 659
 bitmap, 95, 444, 450
 bkmk, 276
 blatant, 579
 blazing, 276
 blend, 798
 blind, 386
 BlobStorageTargetState, 842
 blockdev, 196
 blogdoezequiel, 56
 blogging, 84
 Boolean, 262, 420
 boost, 58, 123
 Boot Page, 95
 bootstrap, 914
 bootup, 436
 Border Gateway, 615
 bottleneck, 343, 502, 509, 693, 720, 747, 791, 792
 Bourne, 192
 bps, 817
 brass, 125
 breach, 451
 Brendan Burns, 127
 brevity, 268, 293, 616
 bridge, 55, 622, 624
 brownout, 71
 Bruce Schneier, 610
 Brute, 610, 611
 Btree, 747
 Bucket, 770, 771
 buckle, 435
 Buffer Cache, 359, 397
 Bulk Changed, 95
 Bulk Copy, 8

Bulkhead, 901
 bumpers, 677
 burdensome, 541
 busiest, 363
 Business Acceptance, 566
 buzzword, 910, 911
 BxlServer, 179
 BYOL, 151

C

cacheobjtype, 333, 718
 caches, 54, 116, 119
 Caching, 54, 56, 265
 cardinal, 711
 Cardinality Estimation, 219
 carte, 43
 carve, 37, 66
 CatalogCollation, 828
 Catalogs, 31, 393
 catastrophic, 65
 Category Details, 51
 CategoryId, 271
 cautionary, 449, 457, 552, 701
 ceiling, 901-902
 centerincident, 844
 Central, 66, 187, 192, 269, 416-417, 419, 469, 886-887
 Central European, 250
 Central Management, 28-29, 416, 808
 Central Processing, 53, 58-59, 114
 Centralized, 66, 177, 191, 345, 390, 395, 635, 786, 877
 Centralized Management, 29
 chalked, 360
 Change Data, 226
 Change Database, 186
 Change Tracking, 226, 775
 changeable, 724
 changedowner, 583
 changeover, 456
 Check Database, 405
 Check For, 26
 circuit, 59, 901

Circuit Breaker, 901
 circumstance, 421
 circumvent, 122
 ciscopress, 615
 claimName, 135
 clarify, 51, 461, 472, 829
 Classless Interdomain, 649
 Cleanup Mode, 729
 Click Ctrl, 702
 Click Tasks, 635
 Click Tools, 34
 clickheavy, 431
 ClickStream, 923-924
 clipboard, 350
 closure, 227, 286
 Cloud Service, 861
 Cloud Shell, 828
 CloudBlobContainer, 848
 cloudblogs, 10, 206, 521, 622, 793
 Cloudera, 166
 clouds, 779
 clumsily, 749
 clumsy, 372
 CLUSAPI, 338, 533
 Cluster Features, 913
 Cluster Instance, 126
 Cluster Instances, 485
 Cluster Manager, 504, 522
 Cluster Name, 497
 Cluster Shared, 481
 Clustered Index, 707, 711
 Clustering Clustering, 73
 clusterName, 527
 ClusterNode, 506
 ClusterParameter, 519
 ClusterResource, 519
 Cluter Instances, 179
 clutter, 182-183
 Code Page, 225
 Code Snippets, 32-33
 Code User, 430
 codebase, 44
 coding, 45, 102, 608, 668, 680, 880
 coincidentally, 435

coincides, 612
 coinciding, 76
 coined, 125
 collapse, 40
 collapsible, 328, 677
 Collation Collations, 224
 Collation With, 245
 CollationName, 827
 CollationTest, 246
 collision, 258, 610, 901-902
 collocated, 820
 colloquially, 61, 64, 746
 colocation, 526, 528
 Column Encryption, 633-634
 Column Master, 633
 Column Selection, 635
 ColumnEncryption, 633
 columnName, 259
 Columnstore, 276, 325, 401, 742, 761-763
 Columnstore Index, 707, 769
 combat, 738
 Command Line, 5, 14, 543
 Command Prompt, 145-146, 169, 192, 345, 423
 commenced, 101
 commensurate, 790
 CommentsChanged, 303
 commercial, 189-190, 193, 616, 928
 commitlatency, 68
 commitment, 434, 779, 832
 Committed Snapshot, 229, 515
 Committed Version, 660
 Common Internet, 67
 Common Language, 95, 116, 252, 809
 Common Setup, 169-170
 commonplace, 11
 Compatibility Level, 217, 226
 Compilation Much, 734
 compilations, 757
 Compiled True, 31
 compliance, 147, 275, 420-421, 793, 839, 854, 879, 896-897
 compliant, 648, 862, 898

- comply, 282
- composed, 86, 909
- composite, 261, 710
- compound, 63, 261-262, 709, 742, 745, 748
- comprehend, 796
- Compress Backup, 411
- CompressedFlag, 736
- CompressionOption, 427
- Compute Scalar, 737
- concise, 19
- concludes, 529
- concocted, 551
- condensed, 547
- conducive, 424
- confidence, 883
- confident, 749, 799, 825
- Confidential, 607
- configname, 918
- configurability, 413
- Configuration Checker, 5
- Configuration File, 168-170
- Configuration If, 176
- Configuration Manager, 23-24, 70, 164-165, 171, 185-187, 192, 505, 620
- Configuration Settings, 159, 420
- Configuration Wizard, 382
- ConfigurationFile, 168, 170, 489
- Configure Management, 20
- Configure Transport, 206
- configurestreaming, 877
- Configuring Advanced, 844
- Configuring Always, 636
- Configuring Database, 544
- Configuring Minikube, 131
- Configuring Transparent, 195
- configvalue, 920
- conflate, 615
- conflicts, 280, 683, 685, 687-688
- congestion, 890
- Connect Item, 228
- Connection Strings, 891
- Connection The, 601
- Connection Timeout, 891
- connectionless, 613
- Connections Diagram, 865
- connectivityarchitecture, 864
- connector, 69, 166, 385, 916
- connects, 192, 603, 878, 910
- ConnectVia, 897
- Consequently, 808
- Console, 15, 149, 164, 309, 345-346, 366-367, 424, 426, 430, 781
- consolidated, 828
- consolidating, 85, 90
- consolidation, 64, 77, 807
- Consortium, 253
- constant, 265, 437
- Contain Owner, 31
- Contain Schema, 31
- Contained Users, 544, 569
- Contained Windows, 569
- containerized, 909
- containerName, 848, 892-894
- containerPort, 135
- Containment Type, 226
- Contains Equals, 31
- contemporary, 483
- contention, 5, 60, 79, 86, 93, 111-113, 337, 341-342, 372, 747
- Context New, 893
- contexts, 231, 601
- contiguous, 32, 93, 195, 444, 664
- continental, 815
- Control All, 837
- Control Panel, 16
- Control Server, 919
- controversy, 56
- convenience, 416, 444, 458, 564, 650
- convenient, 202, 312, 550, 564, 615, 714, 732, 793, 833
- CONVENTIONAL, 181
- conventions, 411, 420-421, 567
- Conversely, 259, 446
- conversion, 11, 245-246, 260, 704
- cooperates, 661
- coordinate, 70, 252-254, 428, 498, 677
- Coordinated Universal, 841
- coordination, 215, 402
- Coordinator, 30, 500
- Copy Program, 13
- CopyDatabaseName, 847
- Copying, 215, 508
- copying, 50, 163, 214-215, 219, 222, 225, 256, 472, 491, 774
- CopyOnly, 894
- cornerstone, 934
- Cost Threshold, 115, 339, 720
- costbased, 722
- costliest, 708, 757
- costly, 89, 314, 471, 704, 709-710, 724, 738
- costs, 64, 72, 86, 93, 114-115, 132, 151, 470, 695, 704, 791, 807-808, 810, 812, 854, 876, 896, 918
- counsel, 605
- Counter Instance, 42
- Countermeasures, 610
- countermeasures, 610
- counterparts, 197
- counterpoint, 86
- counters, 38, 42, 52, 315, 358-361, 363-367, 395, 397, 501, 531, 797
- counting, 355, 615, 738-739
- Countries, 262
- CountryName, 262
- counts, 61, 86, 209, 319, 336, 350-352, 355-356, 710, 714, 756
- couple, 43, 353, 476, 857, 885
- Create Audit, 642
- Create Availability, 858
- Create Catalog, 184
- Create Computer, 497
- Create Endpoint, 858
- Create External, 919
- Create New, 826
- Create Or, 714
- Create To, 348
- CREATETSQL, 597-599

- Creating Microsoft, 371
 - CreationTime, 428
 - Cred, 823, 847-848
 - Cred In, 823
 - Credentials Are, 187
 - Credit, 525, 616, 631, 638, 861
 - criteria, 112, 596, 669, 690, 709, 713, 748, 799, 854
 - criticizing, 794
 - Cross Database, 7, 583
 - crosshairs, 730
 - crossplatform, 345
 - crowded, 414
 - crushing, 677
 - cryptic, 337
 - Cryptographic, 609-611, 613, 617, 620-621, 625, 629
 - Cryptographic Providers, 30
 - cryptographically, 81, 607, 610, 615, 625-626
 - Cumulative Update, 452
 - Cumulative Updates, 200, 377-378
 - cure, 554, 556, 559
 - curious, 129, 637
 - curl, 199, 871
 - currency, 246
 - Current User, 635
 - CurrentControlSet, 898
 - cursor, 33, 110, 245, 311, 430-431, 694, 710, 777
 - Custom String, 639
 - customerfacing, 565
 - CustomerID, 267, 319, 738, 753-754
 - CustomerId, 697
 - CustomerName, 267
 - CustomerNameType, 267
 - Customers, 51, 105, 128, 267, 378, 521, 563, 637, 650, 697, 738, 808, 814, 886, 908, 923
 - CustomerTransactions, 646-647
 - CustomFields, 255
 - customizability, 401, 480
 - customizable, 34, 45, 351-352, 484, 809, 812, 843
 - CycleCount, 266
 - Cyrillic, 245
- D**
- Data Clusters, 127, 905, 907, 909, 911-913
 - Data Collection, 20-22, 30
 - Data Collector, 357, 362
 - Data Compare, 214
 - Data Compression, 320, 762
 - Data Control, 832
 - Data Definition, 398, 533, 578, 678, 695
 - Data Directories, 235
 - Data Discovery, 607
 - Data Disk, 368
 - Data Encryption, 440, 631
 - Data Factory, 804-805, 898
 - Data File, 36, 39
 - Data Integration, 896
 - Data Lake, 746, 817, 914
 - Data Manipulation, 271, 578, 695, 846
 - Data Migra, 882
 - Data Migration, 4, 6-8, 152, 869, 886, 891, 900
 - Data Movement, 919
 - Data Platform, 796
 - Data Portal, 367
 - Data Profile, 51
 - Data Profiling, 51
 - Data Protection, 622
 - Data Quality, 11-12, 207
 - Data Science, 926
 - Data Scientist, 923
 - Data Source, 633
 - Data Storage, 18, 349
 - Data Studio, 3, 349
 - Data Tools, 4
 - Data Warehouse, 20, 25, 44, 314, 607, 816, 908
 - Data Wizard, 50
 - Database Active, 39
 - Database Administrator, 210
 - Database Advanced, 640
 - Database Advisor, 728
 - Database After, 825
 - Database All, 639
 - Database An, 664
 - Database As, 832
 - Database Audit, 646
 - Database Auditing, 641
 - Database Azure, 84, 882
 - Database Backup, 410
 - Database Backups, 8
 - Database Benchmark, 794
 - Database Chapter, 469
 - Database Configuration, 186
 - Database Database, 882
 - Database Diagrams, 34
 - Database Edge, 206-207, 614, 813
 - Database Encryption, 622
 - Database Engines, 28
 - Database Export, 849
 - Database Figure, 824
 - Database For, 6
 - Database Mail, 30, 40, 176-177, 186, 205, 381-386, 401, 535
 - Database Maintenance, 388
 - Database Managed, 7, 150, 382, 471, 886
 - Database Master, 629
 - Database Migra, 882
 - Database Name, 20-21
 - Database Premium, 761
 - Database Properties, 221, 223-224, 230, 233
 - Database Provisioning, 819
 - Database Read, 230
 - Database Recovery, 318
 - Database Schemas, 884
 - Database Security, 832
 - Database Settings, 210-211
 - Database State, 76
 - Database Stretch, 812
 - Database Task, 400
 - Database Threat, 648
 - Database Transaction, 796, 803
 - Database Upgrade, 219, 738

- Database User, 545, 570, 592-593
- databaseCopyName, 847-848
- DatabaseDtuMax, 830-831
- DatabaseDtuMin, 830-831
- databaseelastic, 856
- DatabaseFileName, 326
- DatabaseID, 329
- databaselevel, 217, 226, 835
- DatabaseMail, 382
- databasemailprofile, 384
- DatabaseMailUsersRole, 383
- DatabaseObject, 427
- DatabaseOne, 728
- Databases Databases, 473
- DatabaseUsers, 568
- Databricks, 817, 897, 907, 910
- Databricks Notebook, 896-897
- datagram, 613
- datareader, 572-573
- dataset, 11, 18, 106, 351-352, 676, 774
- datastores, 897
- datatype, 373, 635, 764, 889
- datawriter, 572-573
- Date Created, 409
- Date Equals, 31
- Date Modified, 409
- Date Time, 42
- datecapture, 335-336
- dated, 363, 377
- datetimeoffset, 250-251, 335, 743
- dating, 523
- daunting, 658
- David, 285
- David Klee, 86
- davidklee, 86
- Deadlock Graph, 353, 355
- DeadlockID, 354
- deadlocks, 19, 353, 372, 664, 689
- dealbreaker, 500
- Dealing, 15, 226, 241, 378, 657, 661, 733, 920
- deallocated, 309, 579, 793
- deals, 122, 124, 146, 342, 471, 606, 707
- debate, 57, 287, 564
- debugger, 19, 348, 352, 632
- decade, 284, 347, 422, 476, 615-616
- decipher, 881
- decisionmakers, 607
- decompression, 107, 763
- decrease, 63, 215, 361, 610, 694, 852, 854, 900
- Dedicated Admin, 201
- Dedicated Administrator, 113, 601
- deduce, 611
- Default Profiler, 205
- Default Setting, 430
- Default Snapshot, 479
- defaulted, 392
- defeated, 521, 609-611, 621
- defect, 417-418
- defend, 59, 373, 451, 605, 607-609, 619, 627, 640, 653
- deferred, 465, 734
- Deficiency, 882
- Definition Shared, 664
- defragmentation, 40, 458
- defragmenting, 458
- degradation, 19, 233, 361, 369, 874, 883
- degradations, 874
- degrade, 157, 409, 883
- Delay Between, 395
- delegate, 82-83, 485, 557, 864, 917
- deliberate, 567
- DeliveryInstructions, 303
- DeliveryInstructionsChanged, 303
- delta, 96, 325, 450, 688, 749
- deltastore, 765
- delve, 607, 782, 819, 905
- demostorage, 892
- Demote, 507, 527-528
- denote, 258, 370
- denydatareader, 573
- denydatawriter, 573
- denying, 561
- Deploying Big, 905, 912-913
- Deployment Depending, 933
- DeploymentMode, 162
- Deprecated Features, 52
- deprecating, 7, 52, 518, 871
- descendant, 258-259
- Descendants, 286
- descendants, 259, 286
- descending, 263, 719, 750
- Description Unattended, 170
- descriptions, 27, 147, 161, 405, 424, 479, 804, 881-882
- descriptive, 29
- deserializers, 910
- deserve, 706, 758
- designate, 13, 43, 416, 486, 836
- desirable, 282, 564, 582, 744, 788
- Despite, 49, 224, 251, 307, 734
- destabilize, 931
- Destination Server, 8
- DestinationSqlCredential, 895
- destructive, 147, 319, 323
- DETACH, 123, 215, 220, 236-237
- Detached Cluster, 489
- Detaching, 215, 236-237
- DetailId, 278-279
- determinant, 518
- determination, 883
- determining, 230, 316, 569, 863
- Deterministic, 269, 631-632, 635
- detriment, 691
- detrimental, 58, 280, 678
- Development Servers, 28
- deviation, 51, 223
- DevOps, 125, 137
- diag, 877
- diagnose, 38, 663
- Diagnosing, 19
- diagnosis, 358
- diagram, 48, 291, 909
- dials, 339
- dictate, 59, 113, 457, 791, 821
- Different Location, 8
- Different Server, 419
- Differential Changed, 95
- differentiate, 355, 832

differentiates, 72, 85, 192
 Differentiating, 484
 differentiation, 800
 differently, 232, 298, 362, 375, 401, 667, 749, 762, 858
 differing, 62, 660
 differs, 49, 200, 221
 Direct Access, 68-69
 DirectAttached, 787
 Directory Admin, 719
 Directory Authentication, 549, 837
 Directory Azure, 83
 Directory Database, 859
 directoryadministrator, 878
 dirtied, 232
 Dirty Read, 659
 disadvantage, 66, 401, 617, 746
 disallowed, 278, 546
 disappear, 291, 615
 disapprove, 29
 disassociate, 215
 Disaster, 72, 435, 471
 Disaster Recovery, 484
 disasters, 433, 451, 845, 876
 disc, 151
 discarded, 59, 447, 650
 Discarding, 251
 disconnect, 212, 270, 321, 503, 535, 744
 discontinue, 3, 52, 161, 240, 782
 Discontinued Database, 7
 discontinuing, 52
 Discourages, 877
 discrepancies, 879
 discrete, 77, 89, 145, 594
 Disk Defragmenter, 313
 Disk Only, 74
 Disk Transfers, 363
 Disk Usage, 326, 328
 diskadmin, 558
 disparate, 904
 Docker Desktop, 128, 131
 Dockerfile, 86
 Does Activity, 40

Does Azure, 43
 Does Not, 31
 dollars, 435, 678
 Domain Controller, 5
 Domain Controllers, 87
 Domain Name, 613
 Domain Service, 80, 82, 810, 877
 dotted, 700
 downfalls, 557
 downgrade, 148, 790, 851, 876
 downside, 86, 97, 287, 441, 800
 downtown, 436
 downward, 118, 364
 dows, 882
 drag, 402, 413
 dramatic, 110, 498
 drastic, 74, 219, 226
 drastically, 197
 draw, 257, 287, 370, 476, 480, 680, 707, 761, 858, 869
 drawn, 254
 dreaded, 396
 dried, 797
 drift, 923
 drill, 368, 371, 469, 850, 883
 Drop Availability, 856
 Drop Database, 237
 DS, 636, 792, 810
 During Azure, 825
 Dynamic Management, 556
 dynamically, 98, 118, 132-133, 485, 707, 765, 772, 874

E

Elastic Job, 808-809
 Elastic Scale, 809
 Elastic Transactions, 859
 elastically, 896
 Elasticity, 780
 ElasticPoolName, 828, 830-831, 847
 electrocuting, 71
 Electron, 44, 467, 614, 616, 618
 ElevatedRights, 589
 elevating, 252, 671
 ellipsoidal, 252
 Email Report, 412
 Email Settings, 186
 emailing, 42
 emerge, 287
 emergencies, 505
 Enable Audit, 643
 Enable Optimize, 180
 Enable Worker, 163
 Enabled, 420, 633, 840
 Enabled If, 898
 Enabled No, 41
 EnableHttpsTrafficOnly, 840
 Enabling Forced, 120
 Enabling Service, 386
 encapsulate, 371
 encapsulated, 720
 enclave, 632, 634, 636-637
 EnclaveProviders, 636
 enclaves, 631-632, 636-637
 enclose, 425
 Encryption Data, 611
 Encryption Keys, 187
 Encryption Let, 220
 encrypts, 623, 627, 813, 879
 Enforce Password, 550
 EnforcePasswordPolicy, 549
 Engine Functionality, 7
 Engine Service, 154, 171
 Engine Tuning, 16
 enlightenment, 64, 68
 enormous, 816
 enrichment, 11
 ensuing, 119
 Enterprise Agreement, 861
 Enterprise Core, 203
 Enterprise Dev, 861
 Enterprise Edition, 473, 483
 Enterprise Enterprise, 473
 Enterprise Key, 390
 Enterprise Linux, 5, 128, 193, 491, 493, 522
 Enterprise Security, 542

Enterprise Standard, 473
 enterprisellevel, 229
 entity, 84, 227, 260, 281-282, 287,
 290-291, 297-298, 539, 544,
 569, 618, 666, 694
 Entity Framework, 257
 envision, 130
 eNVM, 861
 ephemeral, 88, 132, 744, 765
 Eponymous, 762, 775
 Equal, 31, 42, 157, 224, 228, 386, 509,
 712, 829
 Equal Between, 31
 Equal More, 31
 Error Log, 35-36, 103, 120, 182, 205,
 238
 Error Number, 42
 errored, 393, 883
 Evaluate Mode, 421
 Evaluate Policies, 30, 421
 Evaluation Edition, 148
 Evaluation Mode, 420
 evaluations, 420-421
 Event File, 351
 Event Hub, 839
 Event Log, 239
 Event Time, 647
 Event Tracing, 352
 EventDuration, 328
 EventFileTarget, 353
 eventful, 182
 EventGrowth, 328
 Events Files, 351
 Events Session, 357, 562
 exacerbate, 372
 Example Live, 700
 Example Query, 703
 Examples Here, 427
 Excel, 45, 51, 161, 341, 574
 Excel Power, 908
 Exections, 39
 executable, 25, 184, 202, 382, 384,
 797
 Executable Minimum, 383
 executables, 152, 170

exfiltrated, 609, 844
 Expand Stored, 50
 expedition, 562
 expended, 813
 expenditure, 779
 expends, 31
 Expensive Queries, 39
 experimentation, 881
 Experimentation Assistant, 883
 expert, 610
 expertise, 540, 745, 873
 ExpirationDate, 421
 Explain Plan, 695
 explode, 722
 exploit, 609, 694
 exploration, 381
 Export, 13, 27-28, 30, 350, 369, 420,
 436, 846-849
 Export Current, 420
 Export Wizard, 51
 ExportRequest, 848
 exportRequest, 848
 exportStatus, 848
 Extended Event, 315, 328, 352
 Extended Events, 16-17, 30, 328, 806
 Extended Support, 379
 ExtendedPrice, 270, 766-767
 Extending, 47, 83, 310, 523
 extensibility, 10, 45, 933
 ExtensibilityData, 176
 extensible, 34, 44, 624-625
 Extensible Key, 622
 External Data, 166, 921
 External Tables, 920
 ExternalPurchaseOrder, 921
 extrapolates, 370
 extras, 616

F

FacetDomainControllerCheck, 5
 Facets, 5, 176, 382, 602
 FactorsThatDelayTruncation, 100
 Factory, 804-805, 834, 855-857,
 896-899

Failover Availability, 473
 Failover Clustering, 30, 73-75, 77-79,
 431, 485
 Failover Clusters, 423
 FailoverClusters, 506, 519
 FailoverPartner, 520
 failurehappen, 745
 FClevel, 505
 fdhost, 774
 feasibility, 68, 319, 445, 463, 512,
 712
 feats, 801
 Feature Pack, 345
 fellow, 546
 Fibre Channel, 67
 fickle, 467
 fictional, 373, 438
 File Allocation, 122
 File Explorer, 192
 File Extensions, 383
 File Header, 95
 File Menu, 430
 File Share, 74, 496
 File System, 50, 67, 122, 145, 239,
 288, 564, 906
 File Transfer, 615
 File Viewer, 647-648
 FileId, 326
 FileLocation, 326
 Fill Factor, 314-315
 Finish Waiting, 41
 FIPS, 898
 FIPSAAlgorithmPolicy, 898
 Firewall Rule, 824
 FirstVal, 265-266
 FixQuorum, 506
 FK, 319, 753-754
 Force Encryption, 620
 Force Plan, 731
 Force Restore, 894
 ForEachObject, 428
 Foreign Key, 749
 Forwarding Pointers, 313
 Fsutil, 145-146

FT, 338
fulltext, 774

G

GameWeek, 757
GameYear, 757
garbage, 289
gauge, 38
gauging, 362
General Data, 606
General Differences, 857
General Distribution, 378
General Purpose, 861, 888
georeplicated, 472
geospatial, 802
Get Service, 917
GetAncestor, 259
GetLevel, 259
GetSmoObject, 178
gettable, 328-329
Getting Started, 344
GetUserName, 278
gigabit, 815
gigabyte, 291, 814, 822
GitHub, 44, 49, 84, 130, 137, 277, 344, 346, 367, 738, 781, 819, 901, 916, 927
githubusercontent, 277, 866, 912
Glacier, 66
glance, 29, 222, 368, 798
Glenn Berry, 119
Global, 18, 384, 502, 509, 662, 821, 860, 878, 887, 896, 925
Global Allocation, 95
Global Fields, 347
Global Rules, 167
Grafana, 910
grail, 137
Graphical User, 883
grasp, 361, 428
grave, 428
GridView, 344, 346-347, 430
Group Policy, 83
Group Wizard, 521, 524

grp, 374
GTH, 890
GUID, 145, 257-258
Guides An, 724
GUIDs, 257-258

H

hacluster, 526-527
Hadoop Distributed, 906
Hadoop Spark, 928
hallengren, 467
halves, 74
Hamilton, 285
handcuffed, 218
handful, 332, 718, 748
handshake, 621
hang, 688
haphazardly, 749
harbinger, 396
hardcoding, 517
harden, 533-534
HardMemoryLimit, 173
Hardware Security, 623
hardwaresecrets, 194
harm, 111, 572
harming, 735
hasten, 557
Haswell, 861
Hat Enterprise, 200
haystack, 766, 770
HBase, 907
Head Blocker, 38
Hekaton, 205
hellish, 437
Help Backup, 427
Help Invoke, 425
helptext, 34
hertz, 58
heuristics, 734
Hierarchyid, 252, 258-259, 281, 802, 811, 889
High Availability, 30, 414, 483, 872, 913

High Performance, 61, 178-179, 194, 787
hijacked, 615
Histogram, 348, 351, 355-356, 711
History Chapter, 182
History Cleanup, 409
Hit, 119, 168, 359, 397
Hit Ratio, 359, 397
hiveQL, 908
Hoc Workloads, 119, 180
homogenous, 887
honored, 217, 597
hood, 8, 614
horribly, 709
Hortonworks, 920
Hortonworks Hadoop, 166
Host Guardian, 636
Host Name, 38, 347, 351, 372, 523
hotkeys, 34
hotspot, 93
hover, 697, 707
How Kerberos, 82
htop, 366
HttpsCertThumbprint, 163
hub, 337, 379, 786
huge, 614, 733, 783
Huge Pages, 195
hugely, 206, 616
Hyperscale, 799-802
Hypertext Markup, 616
Hypertext Transport, 616
hyperthreaded, 203
hyperthreading, 157, 768
hypothetical, 17

I

ICMP, 613
idempotent, 600
ietf, 613-614, 621
If Transparent, 440
ifconfig, 423
Ignore Replica, 415
Impala, 910

imperfect, 712
 impetus, 860
 Implied, 750
 Import Policies, 30
 Import Registered, 29
 inability, 281, 467
 inadvertently, 212, 223, 281
 inadvisable, 668
 inbound, 543, 785-786, 863-864, 872
 inbox, 177, 395
 incapacitated, 438
 incase, 875
 inching, 694
 Include Actual, 697
 Include Step, 394
 incoming, 112, 650
 incompatibility, 762
 inconsequential, 739
 incorporating, 258
 Independent Disks, 65
 Index Rebuild, 313, 408, 746
 Index Reorganize, 408
 Index Scan, 699, 766
 Index Seek, 707, 710, 752-754, 768
 IndexName, 760
 indiscriminately, 329
 inexact, 712
 infoschema, 890
 infrequent, 87, 479, 754
 ingest, 911
 inherent, 284, 412, 631, 742
 iniSetup, 171
 Initialize Secondary, 475
 ink, 797
 inline, 637
 Inner Join, 698
 innumerable, 70
 InProgress, 848
 Input, 62, 609, 709, 733
 Insert Snippets, 33
 insignificant, 123
 insists, 227
 instability, 119
 Install Based, 169

Install Extension, 47
 installaProtocol, 171
 Installation Center, 3-4, 6, 9, 22
 Instance Collation, 141, 154
 Instance Feature, 165
 Instance Should, 473
 instanceconfigure, 867-868
 InstanceNameTem, 186
 Instancepath, 348, 351, 533
 instancetransact, 860
 Instant File, 14, 154
 Instantaneous, 801
 instantly, 105-106, 123, 206
 instruments, 125, 565
 insulate, 56
 insurance, 72
 intact, 312, 416, 446, 584
 IntegerData, 328
 integratedConsole, 431
 integrates, 478, 817, 874
 IntelliSense, 13, 32-33
 interchangeably, 547
 Interconnect Express, 63
 Internetconnected, 426
 internethistorypodcast, 616
 interpreted, 245, 277, 616, 710
 interrogating, 315
 interruption, 491, 507
 intersections, 253
 intimate, 438
 Intranet, 885, 900
 intrepid, 438
 intricacies, 464, 487
 intrinsic, 737
 intrinsically, 277
 introspection, 371
 intrusion, 396
 intuitive, 180, 183, 392
 invoiced, 738
 InvoiceLineID, 750, 766-767
 invoked, 637
 iostat, 367
 IPAddr, 528
 ipconfig, 423

iPhone, 616
 IPsec, 69
 ipynb, 916
 IR, 896-897, 899
 ironclad, 679
 ironic, 745
 isolate, 163, 403, 607, 631, 658
 Istio, 128
 IsValid, 255
 Italian, 203
 Itaniumarchitecture, 927
 iterate, 326
 iterative, 374

J

Jakobsen, 285
 Job Activity, 41, 391
 Job Category, 42
 Job History, 183, 392-393
 Job Notifications, 40
 Job Properties, 390
 Job Reporting, 399
 Job Runnable, 42
 Job Step, 388-389, 394
 jobhistory, 183, 409
 jobsteplog, 394
 jumpstart, 658
 Jupiter, 48
 Jupyter, 48-49

K

KBytes, 365
 Kernel, 48, 56, 189-190, 194-196
 Kestrel, 927
 Key Lookup, 710, 752-754
 Key Range, 664
 keyed, 709
 keystrokes, 34, 71
 keytab, 206
 kilobytes, 175, 531
 Kind Storage, 840
 knockon, 467
 Korea Central, 785

Korean, 203
 Korn, 192
 Kruger, 118
 kubeadm, 912-913
 kubect1, 126, 132, 134, 136, 912
 Kubernetes Although, 131
 Kubernetes Architecture, 127
 Kubernetes Kubernetes, 125
 Kubernetes Once, 132
 Kusto, 367, 370-371
 Kusto Query, 369

L

Lacks, 147, 215, 223
 lag, 89, 530-531
 laid, 437
 lake, 817, 906
 Lake Storage, 817, 915, 922
 land, 141
 landscape, 905
 Language Extensions, 165-166
 Language Statements, 578
 Large Hadron, 483
 Large Object, 890
 Last Run, 42
 LastName, 282, 582
 LastVal, 265-266
 latch, 280, 342, 557, 741, 769, 771
 Launch Pad, 905, 933-935
 launched, 126, 131, 151, 170, 351, 366,
 417, 933-935
 lazy, 78, 97
 leakage, 637
 leaked, 609, 617
 leaks, 605
 leaps, 658
 learned, 531, 787
 Learning Server, 175
 Learning Services, 165
 ledger, 675
 Left Semi, 700
 leftmost, 296, 705
 Legacy Cardinality, 211

Legend, 447
 legislation, 605-606
 legitimate, 120, 122
 LeMaire, 423, 891
 Len, 353
 lengthened, 379
 Level Agreements, 472
 Level Database, 473
 Level Security, 206
 Leveraging Big, 166
 LHC, 483
 liberal, 267
 License Terms, 152
 licenseguide, 127
 life, 184, 281, 358, 363, 378-379, 423,
 565, 659, 677, 782-783, 821,
 826
 lifecycle, 209, 575, 888, 908
 lifespan, 63, 458
 Lightweight Query, 231
 Limit Size, 183
 Limiting Machine, 175
 limitless, 814
 LimitRest, 764
 Line Interface, 150, 781, 908
 linear, 796
 LineProfit, 766-767
 lines, 54, 83, 129, 253, 428, 489, 528,
 694, 700, 707, 848
 link, 6, 9, 70, 150, 152, 167, 186, 239,
 346, 696, 698, 877, 891, 922
 Linked Server, 30, 600, 895
 LinkedIn, 83-84
 Linus Torvalds, 189
 LinuxAG, 525, 528
 Linuxspecific, 194
 ListAvailable, 426
 literacy, 423
 literal, 103, 132, 613, 696, 713, 733
 LiteralPath, 426
 Livy, 915
 Load Balancer, 521
 Local Administrator, 6
 Local Audit, 206

Local Jobs, 417
 Local Machine, 635
 Local Policies, 124
 Local Security, 124, 154, 390
 Local Server, 28-29
 Locale, 107, 206
 Locator, 609, 743
 Lock Granularity, 664
 Lock Mode, 664
 Lock Modes, 664-665
 Lock Pages, 180-181
 Lockable Resources, 664
 Log Analytics, 369-372, 839, 877
 Log Files, 328
 Log On, 16
 Log Reader, 389, 478
 Log Sequence, 434, 437
 Log Shipping, 76, 475
 LogFiles, 386
 Logging Level, 384
 Logical Reads, 39, 730
 Logical Writes, 39, 730
 logicalservername, 841
 Login Properties, 552
 LoginName, 19
 Logins Task, 596
 logistical, 772
 logistics, 307
 logless, 691
 longrunning, 515
 longstanding, 906
 longterm, 854
 LowMemoryLimit, 173
 lumped, 764
 Lunches, 423
 luxury, 414, 785

M

MaC, 258, 744
 Machine Language, 10
 machinery, 234, 494, 565
 magnetic, 61
 magnitude, 54, 114, 249, 276, 469

- Mail, 205, 383-385, 435
- Mailbox, 385
- mailitems, 385
- Mainstream Support, 378-379
- Maintaining Indexes, 317
- Maintenance Cleanup, 409
- Maintenance Plan, 30, 207, 317, 381, 400, 402-410, 412-415, 456
- Maintenance Solution, 467
- Maintenance Task, 124, 213
- maintplan, 409
- malfunctioning, 176, 532
- maligned, 723
- malware, 180
- Manage Categories, 388
- Manage Instances, 876
- manageability, 144, 464, 595, 821
- manageable, 314, 413, 728
- Managed Backups, 812
- Managed Disks, 787
- Managed Lock, 320
- Managed Service, 897
- ManagedComputer, 178
- Management Data, 21-22
- Management Protocol, 613
- Management Studio, 192, 696, 699, 750
- Management System, 657
- managementviews, 316
- Managing Azure, 780
- Managing Spark, 907
- Manning Publications, 423
- mantissa, 248
- Manual Backup, 512
- MapReduce, 906, 910
- MariaDB, 817
- mariadb, 817
- mark, 100, 399, 453, 462, 503, 731
- Mark As, 177, 395
- Markdown, 48-49
- Markup Language, 125
- marts, 606
- masquerade, 73
- Master Boot, 145
- Master Data, 9, 11, 207
- Master Key, 622, 634-635, 922
- Max Degree, 116, 221, 230, 340
- Max Server, 57, 87, 113, 116-119, 159, 366, 715, 787
- Max Size, 729
- Max Worker, 118-119
- maxed, 898
- maximize, 212, 287, 315, 900
- Maximum Buffer, 352
- Maximum Degree, 339
- Maximum Degrees, 157, 310
- Maximum File, 383
- Maximum Server, 159-160
- MaximumMemoryLimit, 175
- MaxLength, 259
- MaxSizeBytes, 828
- MaxVal, 264, 266
- meant, 18, 112, 232, 749, 808
- meantime, 465
- MediaPathLength, 5
- medical, 744
- megabytes, 39
- Mem, 358
- Memory Access, 68
- Memory Allocated, 40
- Memory As, 117
- Memory Consumption, 730
- Memory Grant, 733
- Memory Limit, 705
- Memory Manager, 365
- Memory New, 159
- Memory Pages, 364
- Memory Performance, 365
- Memory Use, 38
- Memory You, 180
- MemoryLimit, 175
- memorylimitmb, 56
- MemoryOptimizedTable, 690
- MemorySafetyMargin, 175
- MemoryThreshold, 175
- Merge Agent, 479
- Merge Audit, 841
- Merge Azure, 809
- Merge Extended, 351
- Merge Join, 698, 700, 710
- Merge Results, 30
- Message Protocol, 613
- messaging, 347, 542, 566
- metaphorically, 565
- metastore, 909
- meter, 25
- MetricName, 369-370
- Metrics, 36, 45, 51, 307, 315, 331, 336, 357-363, 365-372, 700, 703, 730, 814, 877
- microcode, 59, 88
- microprocessors, 59
- microservice, 900-901
- Microsoft Access, 881
- Microsoft Active, 837
- Microsoft Azure, 6, 53, 371, 381, 471, 539, 658, 746, 779, 781, 783, 785, 795, 802, 813, 819
- Microsoft Container, 128
- Microsoft Cryptographic, 623
- Microsoft Data, 217
- Microsoft Distribution, 500
- Microsoft Download, 837
- Microsoft Drivers, 633
- Microsoft Edge, 616
- Microsoft Excel, 11, 187, 423, 438
- Microsoft Exchange, 385
- Microsoft Feature, 50
- Microsoft Hyper, 86
- Microsoft Internet, 616
- Microsoft Knowledge, 621
- Microsoft Knowledgebase, 58
- Microsoft Management, 15, 780
- Microsoft Monitoring, 369
- Microsoft Office, 83, 162, 339, 341, 542, 574, 810
- Microsoft Power, 165, 188, 574, 805
- Microsoft Research, 802
- Microsoft Support, 58, 116, 120, 158
- Microsoft Tech, 842
- Microsoft Windows, 4
- microsoftpresstore, 256
- Microsoftprovided, 521

microsoftsqlserver, 429
 MigraExperimentation, 882
 Migration Assistant, 216-217, 812, 881, 884-885
 Migration Code, 892
 Migration Migration, 882
 Migration Services, 881, 883, 885, 887, 896-899
 Migration Using, 7
 migs, 756-757
 Minimum Log, 98
 Minimum Recovery, 101
 Minimum Server, 160
 MinionWare, 467
 misbehaving, 523
 miscommunication, 483
 Misconfiguration, 71, 518
 Misspellings, 568
 Mixed Extent, 93
 Mixed Mode, 150, 156, 169
 Mixing, 89, 810, 816
 Module Az, 781
 Moisture, 71
 motherboard, 58, 60, 62-64, 71
 mouse, 192, 435
 Moving Azure, 599
 Moving Windows, 594, 597
 Mozilla Firefox, 616
 Multi Server, 417-419
 Multiserver Results, 30
 multisite, 495
 multistep, 341, 661, 674, 687
 multithreaded, 669, 747
 murky, 58
 muscle, 435

N

naïve, 609
 nance, 171
 nanoseconds, 251
 narrower, 63, 698
 Native Client, 167, 519
 nativemode, 185
 necessitate, 746
 neonatal, 250
 Net Send, 394
 Net System, 679
 NetBIOS, 523
 nets, 860
 Netscape Communications, 617
 Netscape Navigator, 616
 Network Address, 614
 Network File, 911
 Network Load, 187
 Network Security, 543, 648-651, 863, 872
 networked, 612
 Networking Provisioning, 872
 New Audit, 642
 New Availability, 524
 New Database, 20, 213
 New Extended, 357, 562
 New Operator, 394
 New Query, 30
 New Session, 17, 349
 Next Scheduled, 42
 noatime, 196
 Node Majority, 487, 496
 NodeWeight, 506
 NoParallelism, 726
 NoRebootPackage, 5
 NORECOMPUTE, 748
 NORECOVERY, 76, 215, 459-461, 474, 512
 Not Available, 363, 365
 Not Between, 31
 noteworthy, 450, 817
 Notify Operator, 412
 Notifying, 42
 notwithstanding, 39
 novice, 381
 nteraction, 233
 ntext, 95, 268, 319, 323, 580, 811
 ntfsinfo, 145
 nuanced, 226, 244, 246
 nuisance, 396, 419
 nullability, 267
 nullable, 254, 261-263, 317
 nutshell, 787

O

OAuth, 83, 916
 obfuscate, 129
 obfuscates, 599
 obfuscation, 619, 638
 Object An, 664
 Object Class, 644
 Object Name, 426, 644
 Object Plan, 724
 Object Plans, 717
 Object Schema, 644
 ObjectID, 333, 718, 8369
 ObjectiveName, 827
 objectives, 400-401, 434
 objectname, 576
 Objects Explorer, 31
 objecttype, 546
 ofband, 450
 offload, 514
 offpremises, 626
 offrow, 248
 Oftentimes, 224, 396, 545
 Ola Hallengren, 467
 omission, 615
 omit, 16, 546, 615
 On Change, 421
 On Demand, 420
 On Failure, 399
 On Linux, 74-75, 113, 192, 387, 395, 425, 625
 On May, 606
 On Microsoft, 62
 On Red, 197, 200
 On Success, 399
 On Target, 418
 On Windows, 40, 75, 191-192, 387, 396, 885
 Once Transaction, 683
 Online Transaction, 116, 761, 796
 Only Routing, 515-516, 520
 Only Standard, 473
 OOM, 206, 276, 457
 Open Database, 12
 Open Geospatial, 253

Open Source, 891
 OpenID, 83
 Opening, 15, 188, 196, 719, 841, 913
 OpenShift, 128
 opensourcerers, 525
 Operating System, 141, 203, 360, 363, 389
 Operations Manager, 369
 OperationStatusLink, 848
 OpEx, 779-780
 Opinions, 439
 Optimization Level, 704-705
 Oracle, 51, 166, 341, 870, 881-882, 888, 908, 911, 922-923
 oracletosql, 888
 OrderDetails, 268
 OrderID, 268, 302-303, 726-727
 origin, 7, 84, 338, 931
 Origin Resource, 931
 OSes, 191, 193, 202, 206
 OSI, 70
 OSS, 44
 Ostensibly, 57, 612
 Other Considerations, 219
 Other Node, 74
 Other Options, 222, 230
 OtherLanguages, 255
 Out Deployment, 187
 outbound, 385, 454, 785-786, 863, 872, 875
 outcome, 59, 98, 382, 388
 outdated, 453, 468
 outlay, 779
 outline, 4, 377, 433
 Outlook, 383
 outperform, 707, 759, 761, 771
 Output In, 394
 Output List, 753
 OutputDirectory, 345
 outstripped, 364
 Overall Progress, 25
 overcome, 286, 372, 414, 803
 overcommit, 57, 87, 89, 177, 395
 overestimate, 771

overhaul, 234
 Overheating, 71
 overkill, 319, 759
 Overriding, 261, 288, 310, 375, 515, 670-672, 683, 727, 731
 overrun, 318
 oversees, 125
 oversight, 218, 575
 overstated, 711
 overtime, 819
 overuse, 342, 720
 overwhelming, 381
 Owing, 412, 618, 641
 Ownership Chaining, 583

P

pace, 233
 Pacemaker, 74-75, 78, 480, 493, 508, 522-528
 Pacemaker Instead, 74
 pacemakerLogin, 527
 Pacific Standard, 273
 Pacific Time, 273
 Package Deployment, 393-394
 Package Manager, 345
 Package Progress, 25
 Packages Patches, 193
 PackageTypeID, 766-767
 packrights, 545
 padding, 639
 Page An, 664
 Page Faults, 364-365
 Page Life, 42, 363, 365, 397
 Page Read, 364
 Page Reads, 364
 Page Splits, 315, 359-360
 Page Verify, 222, 230, 308
 paginated, 803
 pairwise, 287
 panacea, 687
 panels, 26, 423, 700
 panic, 438
 Parallel Query, 115
 parallelization, 339, 888, 934
 paramount, 72, 103, 308
 paranoid, 373
 parentId, 682
 ParentObject, 311
 Parquet, 908, 924
 Partition Table, 145
 passively, 756
 passphrase, 610, 617
 Password Policy, 549
 PasswordPolicyEnforced, 421
 Patterns Products, 193
 Paul, 95, 115, 310, 463, 745
 Paul Randal, 95, 310, 463
 Paul White, 115
 pausable, 322
 Peer, 476-477, 480, 866
 penalizes, 66
 penalty, 60, 65, 116, 434
 pending, 6
 penor, 837
 perceived, 746, 749, 794
 Percentage, 51, 173, 175, 314, 324, 360, 367-369, 406-407, 700, 705, 756-757, 811
 percentile, 369-371
 Percona, 128
 Perform Volume, 124, 213
 Performance Condition, 42
 Performance Monitor, 118, 315, 357-359, 361-364, 366, 395, 397
 performant, 734
 Perimeter, 608, 620
 Period Security, 379
 periodic, 74, 182, 227, 814, 843
 Peripheral Component, 63
 Permission Container, 893
 Permissions Granted, 239
 Permissions Let, 578
 persistence, 98
 Persistent Memory, 691
 Persistent Volume, 127, 132
 pertain, 546, 578, 658, 677, 722
 Pessimistic, 256, 689, 769

- petabytes, 906
- Phantom, 659, 661, 672, 675-677, 680-681, 689-690
- Phantom Read, 659
- phantoms, 663
- phenomena, 660, 663
- philosophical, 190
- phrases, 288, 610-611, 774
- Physical Disk, 368
- Physical Reads, 39, 730
- Physical Sector, 146
- PhysicalDisk, 363
- physically, 70, 87, 238, 290, 318, 323, 444, 565, 751, 822
- pie, 881
- placeholder, 33
- Plaintext, 635
- Plan Correction, 732
- Plan Count, 39
- Plan Found, 705
- Plan Guide, 714, 723, 725-726
- Plan Guides, 723, 725, 727, 730, 732
- Plan Wizard, 403
- planar, 252
- PlanCount, 716
- PlanGuideDB, 725
- PlanGuideName, 725
- Plans Personal, 413
- PlanSize, 718
- PlanUse, 716
- Platform Abstraction, 206, 358, 366, 622
- PlatformCheck, 5
- pleasant, 366
- plethora, 666
- plumbing, 87
- Podcast, 616
- poetically, 260
- Point Counterpoint, 86
- Point Objective, 72
- Policy History, 420
- Policy Management, 30, 420-421
- politically, 554
- Polly Bulkhead, 901
- Polybase, 657, 706
- polygons, 253
- Polymorphism, 287
- Pool Extensions, 801
- portability, 250
- portable, 226, 831
- Portuguese, 203
- possesses, 557
- Possible Owners, 504
- Postmigration, 890
- posts, 281
- posture, 816
- pound, 920
- Power Pivot, 161
- Power Saving, 787
- PowerBI, 188
- powershellgallery, 423
- prebuilt, 420, 908
- preconfigured, 32
- preconnect, 332
- predates, 311
- predecessor, 618, 621
- predefined, 387, 913
- predetermined, 347
- predominant, 339, 747
- prefill, 167
- Premium Assurance, 379
- Premium Storage, 789-791
- Preparing Azure, 434, 819, 832-833, 845, 847, 849, 851, 853
- prepaying, 799
- prepending, 852
- prepopulate, 167, 403
- Prevent Evaluation, 421
- Previous Committed, 659
- Pricing Tier, 888
- Primary Key, 745, 763, 769-770, 775
- Primary Replica, 533
- Private Link, 786
- Probes, 344, 346
- Procedure Cache, 715-717, 729
- Process Explorer, 180
- processadmin, 553, 556
- Processor Affinity, 122
- Processor Time, 37, 358
- ProcessUtilization, 360
- ProductId, 268, 271
- productive, 32
- ProductName, 271
- ProductVersion, 172
- Profile Viewer, 52
- profound, 660
- Prohibited Attachment, 383
- prohibition, 621
- Project Deployment, 162, 184, 393
- Project Hekaton, 741
- Project Jupyter, 48
- proliferation, 905
- prone, 63, 80, 281, 438, 466, 489, 834
- pronounced, 48
- Properties Window, 697, 703
- Protecting Azure, 639
- Protection Regulation, 606
- protections, 607, 619, 640, 688-689
- protects, 480, 619-620, 622, 631, 634
- prototyping, 926
- provable, 678
- provisioner, 133
- proxied, 863
- Proxy, 388-390, 403, 418, 475, 863, 866, 909, 927
- prudent, 106, 469
- pseudo, 283, 446
- Pseudonymization, 606
- psexec, 436
- Public Key, 618
- publication, 237, 477-480, 788
- Publisher, 477-479, 872, 878
- Publisher Properties, 479
- Puerto Rico, 254
- punctuation, 245-246
- purchases, 339
- purge, 183, 275, 409
- purged, 295, 297
- push, 436, 477-478, 554, 669, 924
- Push Down, 166, 908, 924
- puzzle, 43, 397

PVC, 132-134

PVS, 105

Python, 10, 191, 202, 859, 907,
931-935**Q**

quad, 88

quadrupling, 247

quality, 11, 51, 184, 363, 467, 566, 727

Quality Assurance, 147, 566

Query Optimizer, 211

Query Performance, 797

Query Plan, 695, 697

Query Processing, 733, 738

Query Profiling, 700

Query Results, 30

Query Shortcuts, 34

Query Statistics, 19, 700-701

Query Tuning, 34, 231, 738

QueryMemoryLimit, 173

QueryPlan, 718

Queue Reader, 389

QuickSessions, 349

quickstart, 932

R

radical, 800

Rainbow, 611

Randal, 745

randolph, 129

Randomized, 631-632, 634-635

Randomized Deterministic, 632

randomly, 65, 212, 603, 626

ransom, 379

ransomware, 180, 472

Raspberry Pi, 811

Read Bytes, 368

Read Committed, 105, 229

Read Only, 222, 729

Read Paul, 745

Read Scale, 806

Readable Yes, 473

Readers, 254, 263, 453, 661, 665,
682, 687

readiness, 6, 886

ReadOneTable, 575

Reads Performance, 364

readscale, 806

realistic, 156, 219, 296, 374, 381, 422

realm, 543, 817, 881, 906

RealUser, 576

reap, 746

reassign, 808

reassociate, 236

reattach, 236

reauthentication, 82

reboot, 425, 491, 502

Rebuild Index, 406-407, 414

Recall, 98, 273, 784, 821, 853

recap, 758

receipt, 18

Recent Expensive, 37, 39, 701

reclaim, 238-240, 313, 330, 406, 746

recognize, 94, 147, 649, 663

recompilation, 158, 191, 730

reconfiguration, 510, 517

reconnect, 518

reconstitute, 312

reconstruct, 610

Recovering, 312, 469

Recovery Interval, 98, 232

Recovery Model, 222

Recovery Point, 53, 72, 434, 484,
846Recovery Time, 53, 72, 102, 154, 402,
434, 470, 484Red Hat, 5, 74, 128, 193, 195, 200,
471, 491, 493, 522-523, 919

redefined, 582-583

redeploy, 388

redesign, 760

Redgate Books, 118

Redundant Array, 65

Redundant Internet, 467

reenable, 596

reencrypted, 627

reenlist, 417

reestablish, 872

reexecute, 555

refine, 596

reflect, 238-240, 257, 275, 322, 514,
704, 707, 826, 883

reformat, 147

refresh, 37, 163, 350-351, 761

Refresh Interval, 350

regedit, 423

Registered Servers, 27-28, 35, 416

Registry, 16, 24, 128, 152, 201, 207,
210, 417-418, 436, 898

Regressed Queries, 730

regression, 130, 218, 370, 732, 761

rehearsal, 468

rehearse, 439

reimaging, 744

reinstall, 162, 915

reiterate, 523

rejection, 173

rejoin, 526

Relational Database, 6, 657

RelationshipType, 282, 285-286

relay, 186, 383, 385

relevance, 313, 483

reliability, 3, 16-17, 19, 21, 455, 613,
852, 906

relied, 744

relieve, 56

Reload, 47

Relocating, 234

rely, 7, 96, 361, 363, 459, 504, 613,
651

remediate, 841

Remote Admin, 602

Remote Dedicated, 176, 178

Remote Desktop, 153, 436, 438, 542

Remote Direct, 68

Remote Scans, 706

Remove Agent, 183

Remove File, 429

Remove Filter, 31

Remove Server, 431

render, 49, 113, 369-370

renew, 918

reopen, 47

reordered, 749, 752

Reorganize Index, 406
 repackaged, 813
 repartition, 711
 repeatability, 167, 595
 Repeatable Read, 659
 repercussions, 502
 Replica Traffic, 516
 Replication Agent, 389
 repopulate, 96
 Report Manager, 24, 164-165, 185
 Report Server, 22, 24, 164-165, 186
 reportingtype, 807
 repositories, 137, 192, 198
 reprinting, 277
 reproduce, 350, 699
 reputable, 552
 requisite, 200, 438, 636
 rescheduling, 343, 721
 rescue, 118
 reseeded, 503
 reseller, 521
 resemble, 565
 reservation, 117-118, 326
 Reserved Capacity, 799
 Reset External, 935
 Resiliency, 787, 807, 879, 899-901
 Resilient, 288, 900
 Resilient Buffer, 801
 Resilient File, 124
 resolution, 74, 146, 192, 213, 593
 Resource Monitoring, 307
 Resource Pool, 372
 Resource Waits, 36, 38
 resourceGroups, 871
 resourceintensive, 792
 restorable, 411
 restoration, 474
 Restore Events, 183
 restraint, 854
 resubmit, 613
 resynchronization, 477
 RetentionDays, 428
 RetentionInDays, 840
 rethink, 803

reticent, 115
 retired, 383, 392, 687, 689, 780, 900
 retransmission, 901
 retransmit, 901
 retrievable, 752
 retrying, 386
 reusable, 804
 revamped, 908
 reveal, 59, 326, 502
 Review Maximum, 171
 Review Surface, 171
 revlogin, 597, 599
 revolutionized, 756
 revolves, 307
 rewind, 474
 reworking, 234
 rewrite, 432, 694
 rigging, 487, 505
 rightmost, 296, 705
 rigs, 744
 ripe, 363
 ripple, 900
 rlauncher, 175
 rob, 56
 robust, 24, 148, 351, 412, 422
 rocket, 695
 rodents, 71
 Roles Let, 570
 rollbacks, 104-105, 421, 500
 ollout, 218
 rollover, 328, 353, 534
 RootModule, 426
 Rosenhain, 285-286
 rough, 531, 888
 router, 71, 614
 Routing List, 515
 Row Count, 40, 342, 714, 758, 770
 rowgroup, 325, 765
 rudimentary, 36, 339, 475, 565, 809
 Rules Documentation, 6
 ruleset, 346
 Run As, 152, 168, 403
 Run Outcome, 41
 RunTimeCountersPerThread, 699

RunTimeInformation, 699
 rupted, 473

S

safeguard, 584
 Sales Managers, 837
 SalesAudit, 644
 SalesInvoice, 581
 SalesPrice, 271
 SalesSchemaDeny, 581-582
 SalesSchemaRead, 576, 580-581
 salestext, 225
 SampleId, 585, 591
 SampleName, 828
 SampleNewName, 591
 SampleProcedure, 587-590
 SamplesTest, 453
 SampleTable, 585, 590
 SampleView, 586-587, 590
 sandbox, 147, 150
 sanitize, 609
 SAP, 881-882
 saponsqlserver, 501
 savepoint, 278
 Saving, 601
 Scalability, 73, 615, 784, 789, 806, 851
 scanner, 70, 843
 Schedule, 391
 SchedulerMonitorEvent, 360
 Schema This, 665
 schemabound, 591
 Schneier, 610
 Scoped Configuration, 340
 Scoped Logon, 7
 screenshot, 130, 132, 137, 768-769
 Script Session, 348
 ScriptBlock, 867
 scriptUrlBase, 866-867
 scrutinize, 17
 seamless, 166, 913, 916, 930
 seasoned, 381
 Secondary Only, 414
 Secondary Replica, 534

- secondaryinstancename, 476
- secretKeyRef, 135
- secrets, 527, 913
- Sector, 124, 146, 605
- Secure Shell, 613
- Secure Sockets, 69, 163, 383, 617
- secured, 163, 416, 451, 621, 626, 628, 648, 650, 870, 920
- SecureLocation, 626-627
- SecurePassword, 893-894
- securepwd, 170
- SecureString, 893-894
- Securing Azure, 605, 648-649, 651, 653
- Security Center, 816
- Security Identification, 545
- Security Log, 647
- Security Manager, 838
- Security Project, 609
- Security Vulnerabilities, 88
- securityadmin, 557, 573, 575
- securityrelated, 557
- seed, 478, 511
- Seek Predicate, 753
- seeking, 62
- SeekPlan, 704
- seeming, 720
- segment, 103, 444, 764
- selfhosted, 897
- semantic, 288, 808
- Semantic Search, 288
- semantics, 290, 817
- SendGrid, 383
- SensorDataList, 692
- SensorReadings, 692
- sensors, 811, 907
- sentryone, 702
- Separate Schedules, 404
- Sequence Number, 98
- Serial, 62, 405, 722
- Serial Attached, 62
- serialize, 413
- serially, 115, 405, 721
- Server Active, 19
- Server After, 684
- Server Alerts, 177
- Server Annie, 662
- Server Audit, 398, 513, 600, 641, 644-645, 838
- Server Azure, 3, 786, 790
- Server Before, 141
- Server Big, 127
- Server Blog, 622
- Server Browser, 9, 207
- Server Bulk, 916
- Server Cardinality, 226, 771
- Server Connector, 624
- Server Contributor, 838
- Server Core, 190
- Server Credential, 791
- Server Cumulative, 396
- Server Delayed, 692
- Server Denali, 483
- Server Detecting, 307
- Server Docker, 201
- Server Event, 43, 419
- Server Executing, 700
- Server Express, 164, 198, 523, 526
- Server Extended, 701
- Server Failover, 126, 179, 485
- Server Figure, 350, 370, 372
- Server For, 8
- Server From, 712
- Server Groups, 29
- Server If, 688, 706
- Server Import, 50
- Server In, 400, 668
- Server Inside, 658, 738
- Server Installation, 5, 152-153, 169
- Server Installer, 11
- Server It, 312, 660, 925
- Server Job, 399
- Server Jobs, 417
- Server Locks, 678
- Server Log, 814
- Server Login, 545, 570, 593, 598
- Server Logs, 35, 182
- Server Machine, 909
- Server Maintenance, 381, 402-405, 407, 409, 411, 413, 415
- Server Managed, 402, 410, 420, 812-813
- Server Managing, 326
- Server Memory, 365
- Server Message, 64, 290
- Server Microsoft, 126
- Server Migration, 888
- Server Name, 21
- Server Network, 178, 620
- Server Object, 729
- Server Objects, 27, 29-30
- Server Organizations, 623
- Server Page, 364
- Server Performance, 42
- Server Power, 178
- Server Principal, 647
- Server Profiler, 17, 339, 347-349
- Server Properties, 115-116, 118, 120, 160, 210-211
- Server Querying, 368
- Server Roles, 30
- Server Scheduler, 122
- Server Security, 80, 421
- Server Semantic, 288
- Server Service, 15-16, 240
- Server Setting, 308
- Server Standard, 61, 75, 79, 120, 122, 495
- Server Startup, 184
- Server Target, 880
- Server Tiger, 233
- Server Tools, 25
- Server Trace, 183
- Server Understanding, 328, 334, 657, 682, 702, 720
- Server Updated, 378
- Server Upgrade, 4
- Server Using, 381
- Server Windows, 201, 365-366
- Server Wizard, 417
- ServerProtocol, 178
- Servers Wizard, 29
- Service Account, 24, 185

- Service Azure, 886
- Service Blob, 893-894
- Service Broker, 43, 223, 226, 352, 382, 386-387
- Service Consumption, 932
- Service Pack, 15, 57, 60, 68, 291, 301, 378, 452, 631, 641
- Service Principal, 81-82
- Services Catalog, 219
- Services Command, 188
- Services Execution, 165
- Services Installing, 161
- Services Launchpad, 488
- Services Manage, 163
- Services Options, 881
- Services Package, 50, 403
- Services Scale, 163
- Services Service, 163
- Services Starting, 10, 22, 164
- Services The, 11, 162
- servicing, 307, 377-379, 515, 749, 763, 861
- Set Service, 917
- setColumnEncryptionSetting, 633
- Setspn, 82
- Settings On, 430
- Settings That, 420
- settling, 765
- Setup Bootstrap, 143, 166
- Setup Completed, 25, 185
- setupadmin, 558
- SetupCompatibilityCheck, 6
- severely, 111, 147, 157, 487
- Severities, 396
- Severity Level, 42
- Sewell, 423
- SGAM, 95
- shard, 803, 809, 909, 911, 925
- Sharding, 784, 803, 809, 910, 925
- shardlets, 809, 925
- shareable, 48
- Shared Access, 453, 791
- Shared Feature, 165
- Shared Intent, 665
- Shared Memory, 69, 177
- SharedManagementObjects, 345
- Sharepoint Online, 165
- Sharing, 27, 57, 67, 80, 84, 87, 608, 618, 664, 807, 897, 931
- Sharing Cross, 931
- shedding, 901
- sheer, 708
- Shift, 32, 47, 163, 544, 855
- shim, 251
- shine, 105, 762
- Shoot, 74
- Show Advanced, 174
- Show Live, 40
- showstoppers, 7
- shrink, 211, 213, 228, 327, 329-331, 406, 831
- Shrink Database, 406, 831
- signing, 619
- silently, 167, 246, 638
- Silverlight, 9
- Simple Mail, 382
- Simple Talk, 57
- simplified, 141, 164-165, 214, 615, 802, 873
- Simplified Chinese, 203
- simulate, 131, 585
- Since Express, 76
- Single Server, 926
- singlecolumn, 227
- singlevalue, 359
- Singular, 788
- sinking, 435
- sions, 473
- siteto, 869
- sits, 613, 615-616
- sive, 665
- Size Based, 729
- SizeInBytes, 890
- SizeInKB, 890
- SizeInMB, 716
- SizeMb, 760
- skew, 698
- Skip Using, 512
- skipHeader, 922
- skipped, 123, 141, 265, 284, 309, 399
- skips, 337, 669, 848
- Slack, 423
- Slipstream, 378
- slowness, 19, 38
- smalldatetime, 249
- smallint, 249
- smallmoney, 249
- smartphone, 435
- Snapshot Agent, 478
- Snapshot Isolation, 105, 229
- Snapshots, 30, 86, 100, 208, 410, 440, 462, 469, 801
- snippet, 32-33, 170, 265, 311, 667, 681, 708
- snoop, 607
- socrates, 802
- Software Assurance, 148, 521, 810
- sole, 43
- solidstate, 63
- Solutions Perhaps, 371
- Some Linux, 193
- somecompressedindex, 220
- SomeJSON, 255
- SomeTable, 265
- sophisticated, 227, 367, 370, 653
- Source Server, 6
- sourceCred, 893
- sourceDatabase, 892-893
- sourceInstance, 892-893, 895
- SOURCESQLSERVER, 892
- South Africa, 886
- sovereign, 779
- SpaceUsed, 327
- SpaceUsedMB, 326-327
- Sparse Columns, 268
- Spatial, 252-254, 764, 774-775, 812
- Specifies, 129, 263, 299, 314, 383, 414, 434, 449, 517, 663, 922
- Specifies Quiet, 170
- Specify Replicas, 521
- speculative, 58-59
- spheres, 775

- spids, 333
- spill, 55, 110, 709
- spin, 86, 468
- spinlocks, 233
- spools, 340
- spot, 36, 370, 468, 747, 758
- spouse, 285
- sprawl, 255, 553
- spread, 96, 269, 455
- sprowl, 326
- spun, 469
- square, 441, 553
- squawks, 435
- stack, 35, 194, 423, 532, 613, 619-620, 669, 711, 813
- stacking, 9, 120, 197
- staggeringly, 614
- Stale, 502, 659, 704, 772
- Stale Query, 729
- stand, 64, 153, 435, 912
- Standard Login, 595
- Standard Reports, 328
- Standard Storage, 789, 793
- standardization, 11, 167, 416
- Standby Mode, 76
- standing, 125, 553, 659
- standpoint, 707, 743
- stands, 12, 190, 193, 195, 202, 617
- staple, 125
- stark, 366
- Start Menu, 25
- starter, 337
- Starting Setup, 168
- StartingOffset, 146
- StartTime, 328-329
- Startup Parameters, 184, 240
- starve, 57, 807
- State As, 420
- Statement Table, 733
- Statement Task, 405
- StatementStartInObject, 718
- StatementText, 718
- StateProvinces, 254
- Statistics Statistics, 924
- Statistics When, 228
- steady, 77
- steal, 56, 89, 607
- steer, 744
- Stella, 285
- Stella Rosenhain, 286
- Steve Stedman, 460
- steward, 11
- STGeomFromText, 253
- stick, 436
- stickiness, 926
- stipulations, 763
- stmt, 725-727
- stock, 19, 24, 33, 748
- StockItemID, 750-751, 766-768
- stolen, 605
- stomach, 435
- storAcct, 847-848
- Storage Class, 133
- Storage Direct, 64
- Storage Network, 67
- Storage Open, 51
- Storage Spaces, 67, 481, 790
- StorageAccessKey, 848
- storageAccount, 840, 893
- storageAccountName, 840, 847
- StorageAccountResourceId, 840
- storageaccounttype, 133
- StorageClass, 133
- StorageKey, 847-848
- StorageKeyType, 840, 848
- StorageV, 893
- Stored Access, 453
- Stored Procedure, 897
- straight, 432, 733
- strange, 179
- stray, 551
- stream, 77, 105, 153, 352, 370, 474, 502, 509, 616, 720
- strength, 51
- stress, 854
- Stretch, 207, 482, 811-812
- Stretch Database, 276, 786, 810-812
- stretching, 446
- strict, 69, 254, 281, 660
- strictest, 54
- strictly, 13, 26, 292, 529, 855, 879
- strides, 694
- strike, 821
- stringent, 661
- stripe, 146, 788, 855
- strive, 575
- strongpassword, 594
- struck, 462
- struggle, 157, 438, 468, 794
- stub, 22, 207
- stubborn, 378
- stuck, 120, 436, 662
- Studio As, 25
- Studio Filters, 31
- Studio Let, 696
- Studio One, 48
- Studio The, 26, 405
- sub, 273, 285
- Sub Counter, 42
- subclass, 614-615
- subclause, 272-274, 284-285
- subdirectory, 410-411
- subfolder, 143, 209, 211, 386, 389, 420, 427-428
- subheading, 840
- Subject Alternative, 898
- submarine, 145
- submit, 191
- Subnet Managed, 872
- subnetwork, 649
- suboptimal, 214, 244, 745, 752, 803
- Subplan, 403-404, 412-413
- subqueries, 264
- Subscription Settings, 187
- subscriptionId, 866, 871
- SubscriptionName, 782
- subsection, 18, 32, 172, 224, 244, 466, 559, 799
- subset, 85, 299, 493, 557-558, 838
- subsidiary, 49
- substantial, 255, 472, 479
- substitute, 406, 429

subtracting, 249
 subtree, 754
 succeed, 8, 404, 477, 555, 684
 successor, 483, 626
 suffer, 313, 352, 672, 675, 677, 680, 706
 suffice, 544, 660
 suffix, 31, 297, 639
 suggest, 216, 246, 288, 309, 548, 553, 712, 724, 728, 759
 suit, 344, 558, 748, 760, 813
 suite, 331, 361, 423, 613, 653, 762, 775
 summarize, 369
 summation, 580
 summer, 216, 250
 Summer Time, 250
 sunny, 435
 superuser, 193
 supervisory, 744
 SupportViewServer, 558-559
 Suppress, 172, 182-183, 384
 Surface Area, 176, 184, 382, 420, 422, 602
 surfaces, 747
 surge, 71
 surpasses, 662
 surprise, 438, 446
 surrogate, 743
 surveying, 541
 survive, 70, 433, 446, 793
 susceptible, 88, 681
 Suspended, 332, 334, 392, 530, 535
 Suspended Waiting, 41
 suspicious, 879
 sustain, 485-486
 Sustained, 364, 369, 486
 Swagger, 932-933
 swapping, 87
 switchover, 887
 symptom, 365
 Synapse, 607
 synchronouscommit, 505
 synchronously, 691-692
 sysctl, 195-196

SysEndTime, 271
 sysinternals, 180
 sysjobhistory, 392-393
 sysjobs, 391-392
 sysjobstepslogs, 394
 syslanguages, 278
 sysmail, 385-386
 sysmaintplan, 410, 412
 sysprocesses, 332
 SysStartTime, 271
 sysstat, 367
 System Center, 369
 System Configuration, 152, 166
 System Disk, 19
 System Health, 205
 System Managed, 113
 System Message, 42
 System Policies, 420
 systemctl, 204, 524, 526
 SystemHealth, 360
 SystemIdle, 360
 systemversioning, 275
 syststat, 367

T

Table Name, 889
 Table Scans, 706, 746
 TableName, 247-248, 662, 668, 679, 694-695, 713, 760
 TableNamePattern, 247
 tableoption, 14, 247
 tabs, 34
 Tabular, 10, 26, 161-162, 173-174, 805, 907
 tactic, 226
 tailored, 401
 tale, 449
 tamper, 467
 tantalizingly, 680
 Tape, 64, 66, 401, 411, 447, 452, 454, 857
 Target Agent, 381
 Target Details, 421
 Target Multiple, 418
 Target Recovery, 211
 Target Server, 6-8, 42, 358, 365, 418
 targetCred, 894
 targetDatabase, 892, 894
 targeted, 564, 641, 653
 targeting, 45, 402, 884
 targetInstance, 892, 894-895
 targetLogin, 894-895
 targetmi, 892
 targetPort, 136
 targetserver, 418
 Task Force, 613-614
 Task Manager, 36
 Task State, 37
 Tasks On, 154
 tax, 315
 TaxAmount, 766-767
 TaxRate, 766-767
 tear, 780
 teardown, 819
 techcommunity, 167, 608, 842, 891
 technet, 363, 429
 tedious, 168, 489
 tedium, 141
 telemetry, 45, 348, 653, 877
 telnet, 423
 Temp, 167, 179, 426, 429, 734, 892
 TemplateGuide, 727
 Temporary Storage, 144
 temptation, 25
 tempted, 580, 825
 tenant, 165, 521, 542, 878-879, 897
 tend, 63, 70, 396, 610, 792
 Teradata, 166, 908, 911
 Terminal, 48, 430
 terminate, 56, 206, 320, 685
 termination, 540
 terrible, 734
 territories, 254
 tessellation, 775
 testament, 13
 TestDbcreator, 555-556
 TestDrop, 556
 TestDropSa, 556

- testnolock, 679
 - TestPermissions, 581-582
 - TestSysadminDeny, 554-555
 - TestUser, 576
 - tethered, 435
 - Text Search, 5, 774
 - theme, 34
 - thepollyproject, 901
 - thereby, 86, 196, 250, 833, 851
 - therecovery, 102
 - thesaurus, 774
 - thick, 88, 708
 - thickness, 707
 - thin, 87, 206, 793
 - thinly, 791
 - THP, 195
 - Thread, 699
 - thread, 103, 114, 118, 334, 340, 392, 396, 765, 861
 - threaded, 310, 722
 - Threat Protection, 640, 844
 - threats, 59, 640, 653, 844-845
 - throttled, 851
 - throttling, 374, 888
 - tial, 171
 - tication, 570
 - tick, 428
 - ticket, 81-82
 - Ticket Granting, 81
 - tidy, 312-313
 - tie, 309, 544
 - tigertoolbox, 738
 - tightening, 484
 - tightly, 493, 563, 808, 817
 - tilde, 428
 - tiles, 45
 - timechart, 369-370
 - timeframe, 34, 36, 335
 - TimeGenerated, 369-370
 - timeouts, 372, 518-519, 898, 900
 - timesaver, 783
 - timesheet, 565
 - timespan, 330
 - timetravel, 251
 - todatetime, 370
 - Token Replacement, 398
 - tokenization, 398
 - toolbar, 34
 - toolbelt, 332, 707
 - Toolbox, 347, 403, 413, 422, 741
 - tooling, 128, 913
 - toolkit, 25, 44, 48
 - toolset, 25-26
 - topology, 480, 521
 - touch, 657, 671
 - ToUniversalTime, 847
 - Trace Flag, 112, 121, 146, 157, 181-182, 219, 229, 239, 636, 773
 - Trace Flags, 7, 112
 - tracked, 301-302, 367, 485
 - Tracking Page, 357
 - trail, 648
 - training, 334, 759, 912
 - Transaction Coordinator, 500
 - Transaction Log, 91, 410
 - TransactionScope, 680
 - TransactSQL, 265
 - Transfer Logins, 595-596
 - Transfer Protocol, 382
 - transient, 186, 396, 872, 900
 - transition, 146, 519, 861, 874
 - transitioning, 864
 - transitive, 286
 - Transmission Control, 612
 - transmit, 617
 - Transparent, 223, 226, 372, 624, 774, 879
 - Transparent Data, 182, 223, 511, 622, 832, 857, 871
 - transparently, 195
 - transport, 501-502, 532, 534, 620, 930
 - Transport Control, 70
 - Transport Layer, 513, 617, 620, 879
 - Transport Security, 69
 - transportability, 600
 - tremendous, 680, 691, 741
 - triage, 844
 - trick, 333
 - trifecta, 879
 - trigger, 383, 493, 535, 638, 678, 715, 746, 767
 - Trigger Name, 889
 - trillion, 614
 - trim, 56
 - TrimStart, 893-894
 - trip, 436
 - Triple, 448
 - TRIVIAL, 56, 704-705
 - troublesome, 601
 - troubling, 342
 - truncation, 99-100
 - Trusted Platform, 636
 - TrustServerCertificate, 891
 - Tuning Advisor, 17, 232
 - tunnel, 813
 - Tuple Mover, 765
 - tweak, 468, 724
 - Twilio, 383
 - twosynchronous, 526
 - Type Ctrl, 430
 - Type Details, 51
 - Type Disk, 790
 - Type Full, 894
- ## U
- UAT, 147, 566
 - ubiquitous, 32, 178
 - Ubuntu, 5, 74, 190, 193, 199-200, 493, 522, 543, 919
 - Ubuntu Server, 193
 - UC, 262
 - ucs, 533-534
 - UDF, 113, 735, 737
 - UDFs, 113, 735, 737
 - UDP, 526, 612-613
 - UDR, 651, 864, 875
 - UDT, 267
 - UI, 137, 177, 347, 423, 490, 720, 723, 728, 827, 851
 - Uid, 891
 - UIMODE, 490
 - UIs, 728

UNC, 8, 181-182, 454, 479
 undecillion, 614
 underpins, 73
 Undo, 98, 103, 105, 440, 628
 unencrypted, 91, 121, 223, 234, 244,
 280, 446, 464, 490, 629, 638,
 729, 833, 910
 Unicode, 107, 244-246, 784
 Unified, 37, 164, 582, 915
 Uniform Extent, 93
 Uniform Memory, 60, 116
 Uniform Resource, 609
 uninstall, 9, 161, 167, 919
 unintended, 846
 uninterruptible, 435
 uniqueidentifier, 746
 uniquifier, 745
 Universal Authentication, 541
 universe, 905
 Unix, 190, 202
 Unlike Windows, 781
 UNMASK, 701
 unpack, 151
 unpausable, 322
 unravels, 227
 unreadable, 530, 609, 611, 617
 unrecoverable, 396
 Unsafe, 844
 unsentitems, 386
 unsigned, 249
 unsure, 924
 unwanted, 122, 812
 Up Database, 411
 updatable, 480, 875
 Update Services, 152
 Update Statistics, 324, 406, 408, 773
 Update This, 664
 UpdateEnabled, 153, 170
 UpdateSource, 153, 170
 UPDATETEXT, 580
 UPDLOCK, 668, 687-688
 Upgrade Advisor, 4-5
 Upgrade Used, 882
 Upgrade Wizard, 50
 upkeep, 712

Upload, 151, 455, 917
 UPN, 836-837
 UPS, 435, 466
 uptime, 72, 496
 upward, 220, 365, 614, 787
 URLs, 23-24, 454, 513, 516, 744, 892
 USB, 151, 623
 Use Ctrl, 33
 Use Delete, 871
 Use Encrypt, 825
 Use Invoke, 429
 Use Query, 721
 Use Samba, 885
 Use Tags, 150
 UseCount, 718
 User Acceptance, 147, 566
 User Datagram, 612
 User Defined, 362, 735, 873
 User License, 129
 User Maintenance, 404
 User Process, 37
 User Rights, 124
 userdefined, 243, 802, 843
 UserDetails, 278-279
 UserId, 278-279, 891
 Users Password, 935
 usesparse, 268
 Using Azure, 837, 846, 854, 887
 Using Role, 783

V

vacant, 660
 vacuum, 339, 720
 validate, 270, 478, 610, 835
 ValidFrom, 273-274
 valleys, 370
 Value From, 713
 Value Value, 587-588
 valueFrom, 135
 ValueFromProcedure, 587-588, 590
 ValueFromProcedureDynamic,
 588-590
 ValueFromView, 586-587, 590
 varbinary, 95, 247, 252, 287-289, 319,
 323, 717, 763, 775, 788
 variability, 175
 Variable Deferred, 734
 variants, 665
 VehicleTemperatures, 736
 Verbose, 281, 384, 393, 429
 verification, 82, 96, 449, 619, 930
 Verify Backup, 411
 versa, 47, 74, 162, 362, 422, 708, 762,
 796
 verticallyfiltered, 748
 VertiPaq, 161
 VertiPaqMemoryLimit, 173
 VertiPaqPagingPolicy, 173
 Veteran, 176, 333, 476
 vetted, 554
 Via Object, 750
 viable, 383, 707, 775, 932
 vibrational, 71
 vice, 47, 74, 162, 362, 422, 435, 708,
 762, 796
 victim, 354
 View Audit, 647, 840
 View Facets, 420
 View History, 420
 View Target, 349-351
 Viewing Data, 367
 Viewing Performance, 366
 violate, 421, 638
 violation, 379
 Virtual Address, 173
 Virtual Local, 69-70
 Virtual Machine, 85, 867-868
 Virtual Network, 73, 485, 651, 653,
 785, 832
 Virtual Private, 467, 566, 608, 651,
 813
 virtualip, 528-529
 Virtualization, 53, 58-59, 62, 85-89,
 125, 166, 193, 469, 631-632,
 907-908, 916, 921-922
 virtually, 372, 782, 814
 VirtualNetwork, 650
 virtualNetworkName, 866
 virtues, 919

Visual Analytic, 19
 Visual Studio, 34, 44, 49, 51, 153, 345, 424, 426, 430, 636, 861, 929
 visualization, 368
 vms, 150, 791
 vmstat, 367
 VMWare, 86, 89, 744
 VnetSubnetConflictWithIntended Policy, 873
 vNext, 899
 VNN, 73, 75, 485
 volatile, 54-55, 61-62, 64, 97
 Volatile Memory, 63
 Volume Maintenanceinitialization, 171
 Volume Maintenance, 154
 volumeMounts, 135
 vsphere, 89
 vspherenuma, 89
 vulnerabilities, 58-59, 88, 608-609, 640, 879
 Vulnerability, 59, 608, 816, 842-844
 Vulnerability Assessment, 608
 vulnerable, 621

W

Wait Category, 38
 Wait Events, 36, 38
 Wait Resource, 38
 Wait Time, 38
 Wait Type, 38
 walkthrough, 131, 163, 191, 198, 411, 491, 521-523, 637, 842
 WAN, 482, 485, 495-496, 502, 509
 Warehouse, 19-22, 51, 231, 295-296, 580, 736, 756, 792, 817, 877, 905
 Warm, 798-799
 warranted, 472
 wars, 616
 wary, 422, 532
 watch, 25, 341, 348, 350, 373, 533, 709, 768

Watch Live, 350-351
 Web Application, 609
 Web Chapter, 185
 Web Portal, 22-23, 185-186
 Web Service, 22-23, 185
 webbased, 383
 webdata, 923
 webHDFS, 913
 webpage, 26
 webservice, 185
 Website, 4, 186, 575, 609, 618, 692, 782, 797
 WebsiteExecute, 574-575
 welldesigned, 749
 When Azure, 833
 When Optimize, 119
 When Restoring, 76
 When Torvalds, 189
 When Transaction, 674
 Why Virtualize, 86
 widespread, 158, 218, 905
 WideWorldImportersDW, 507
 WideWorldServerCert, 629-630
 wildcard, 353, 425, 699
 wildly, 616
 Windows Active, 81, 539
 Windows Administrator, 563
 Windows Application, 239, 419, 642
 Windows Authenticated, 221
 Windows Authorization, 563
 Windows Certificate, 637
 Windows Command, 168
 Windows Configurations, 5
 Windows Creators, 543
 Windows Data, 207
 Windows Event, 419, 641, 647
 Windows Failover, 504, 522
 Windows Latin, 225
 Windows Management, 6, 43, 177, 395, 425
 Windows Page, 144
 Windows Performance, 368, 396

Windows Security, 642
 Windows Services, 16, 564
 Windows Start, 11, 15, 124
 Windows Subsystem, 543
 Windows System, 426
 Windows Task, 365-366, 387, 435
 Windows Terminal, 192
 Windows Update, 152-153, 170, 172, 200, 795
 Windowsauthenticated, 548, 597
 Windowsbased, 788
 WindowsFeature, 431
 Windowsonly, 430
 WindowsPowerShell, 426
 Winpremise, 882
 wmic, 146
 WmiServiceStateCheck, 6
 woefully, 438
 workarounds, 860
 workbooks, 161
 Worker Manager, 163
 Worker Node, 163
 Worker Thread, 41
 workflow, 44, 885, 910
 workgroup, 489
 Workgroup Cluster, 489
 WorkingSetMaximum, 175
 WorkingSetMinimum, 175
 Workload Group, 38, 373
 Workload Groups, 374-376
 WorkSetMaximum, 175
 Workspace, 369, 371-372, 839, 877, 918, 933
 World Wide, 190, 225, 605, 615
 WorldWidImporters, 309-310, 312
 Write Performance, 363
 wrought, 506
 WS, 83, 225
 WSFC, 480-482, 485, 492-499, 504-505, 508, 525-526, 528
 WSFCs, 489, 496
 wsl, 543

WWI, 312, 330-331, 443, 460-461,
508-511, 520, 627, 629

X

xcolumn, 370
xdl, 355
XELFile, 353-354
XELPath, 353
Xeon, 58
XEvent, 17-18, 348, 836
Xevent Profiler, 348-349
xevents, 348
XFS, 69, 124, 191, 195, 198
XMLHttpRequest, 931

XQuery, 254
XTP, 229, 337, 343, 533, 860

Y

YAML, 125-126, 133-134, 136
YaST, 193
Yellowdog Updater, 193
Yes Automatic, 473
Yes Available, 304
Yes Status, 41
Yes Time, 304
Yes Tracks, 304
yield, 343, 721
YourNewStrong, 131

YourStrong, 129
yum, 193, 197, 199-200, 526-527

Z

Zen, 193
zeroed, 103, 122
zeroing, 123
zeroinitialized, 123
Zip, 249
Zulu Embedded, 10
Zulu Open, 165
ZYpp, 193
zypper, 193, 199-200