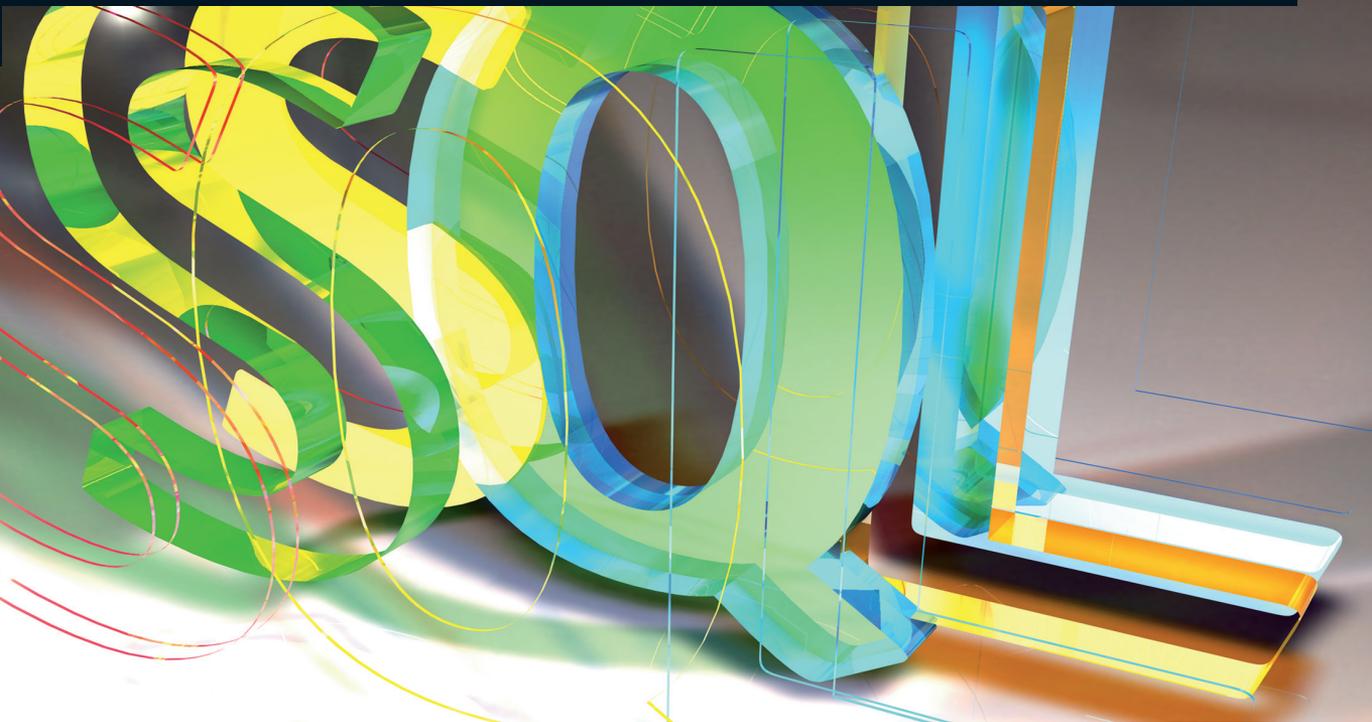


William R. Stanek

Microsoft SQL Server 2012 Ratgeber für Administratoren



Der praktische Begleiter
für die tägliche Arbeit

Microsoft
Press

William R. Stanek

Microsoft SQL Server 2012 –

Ratgeber für Administratoren

Microsoft[®]
Press

William R. Stanek:
Microsoft SQL Server 2012 Administrator's Pocket Consultant
Microsoft Press Deutschland, Konrad-Zuse-Str. 1, 85716 Unterschleißheim
Copyright © 2012 by Microsoft Press Deutschland

Das in diesem Buch enthaltene Programmmaterial ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor, Übersetzer und der Verlag übernehmen folglich keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programmmaterials oder Teilen davon entsteht.

Das Werk einschließlich aller Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die in den Beispielen verwendeten Namen von Firmen, Organisationen, Produkten, Domänen, Personen, Orten, Ereignissen sowie E-Mail-Adressen und Logos sind frei erfunden, soweit nichts anderes angegeben ist. Jede Ähnlichkeit mit tatsächlichen Firmen, Organisationen, Produkten, Domänen, Personen, Orten, Ereignissen, E-Mail-Adressen und Logos ist rein zufällig.

Kommentare und Fragen können Sie gerne an uns richten:
Microsoft Press Deutschland
Konrad-Zuse-Straße 1
D-85716 Unterschleißheim
E-Mail: mspressde@oreilly.de

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
12 11 10

Print-ISBN 978-3-86645-685-3, PDF-E-Book-ISBN 978-3-86645-771-3

© Microsoft Press Deutschland
(ein Unternehmensbereich der Microsoft Deutschland GmbH)
Konrad-Zuse-Str. 1, D-85716 Unterschleißheim
Alle Rechte vorbehalten

Übersetzung und Satz: G&U Language & Publishing Services GmbH, Flensburg (www.GundU.com)
Umschlaggestaltung: Hommer Design GmbH, Haar (www.HommerDesign.com)
Gesamtherstellung: Kösel, Krugzell (www.KoeselBuch.de)

Übersicht

Einleitung	19
Teil I	
Grundlagen von Microsoft SQL Server 2012	27
1 Verwalten von SQL Server-Computern	29
2 Verwalten von SQL Server-Diensten und -Clients	67
Teil II	
Verwaltung und Sicherheit von SQL Server 2012	93
3 Richtliniengestützte Verwaltung	95
4 Konfigurieren und Optimieren von SQL Server-Computern	125
5 Anpassen und Verknüpfen von SQL Server-Computern	147
6 Wesentliche Aufgaben der Datenbankverwaltung	201
7 Sicherheit in SQL Server 2012	259
Teil III	
Datenverwaltung mit SQL Server 2012	311
8 Bearbeiten von Schemas, Tabellen und Sichten	313
9 Indizes, Einschränkungen und Partitionen	359
Teil IV	
Optimierung, Wartung und Wiederherstellung	
von SQL Server 2012	397
10 Automatisierung und Wartung von SQL Server 2012	399
11 Sichern und Wiederherstellen von SQL Server 2012	447
12 Profilerstellung und Überwachung von SQL Server 2012	493
Stichwortverzeichnis	527
Autor und Fachgutachter	543

Inhaltsverzeichnis

Einleitung	19
Für wen ist dieser Ratgeber konzipiert?	20
Wie ist dieses Buch aufgebaut?	21
Was ist SQL Server 2012?	21
Systemvoraussetzungen	22
In diesem Buch verwendete Konventionen	24
Weitere Quellen	24
Support	25
Teil I	
Grundlagen von Microsoft SQL Server 2012	27
1 Verwalten von SQL Server-Computern	29
SQL Server Management Studio	31
Erste Schritte mit SQL Server Management Studio	31
Verbindungen mit einer bestimmten Serverinstanz	33
Verbindung mit einer bestimmten Datenbank	34
Verwalten von SQL Server-Gruppen	35
Einführung in SQL Server-Gruppen und die Ansicht Registrierte Server	35
Erstellen einer Servergruppe	37
Löschen einer Servergruppe	37
Bearbeiten und Verschieben von Servergruppen	38
Hinzufügen von SQL Server-Computern zu einer Gruppe	38
Verwalten von Servern	38
Registrieren eines verbundenen Servers	39
Registrieren eines neuen Servers in der Ansicht Registrierte Server	40
Registrieren zuvor registrierter SQL Server 2000-Computer	41
Aktualisieren der Registrierungsinformationen für lokale Server	41
Kopieren von Servergruppen und lokalen Registrierungsinformationen	42
Bearbeiten der Registrierungseigenschaften	44
Verbinden mit einem Server	44
Trennen der Verbindung zu einem Server	44
Verschieben eines Servers in eine andere Gruppe	44
Löschen einer Serverregistrierung	45
Verwenden von Windows PowerShell zur Verwaltung von SQL Server	45
Starten, Beenden und Konfigurieren des SQL Server-Agents	51
Starten, Beenden und Konfigurieren von Microsoft Distributed Transaction Coordinator	51

Einrichten des Startverhaltens von SQL Server	52
Einrichten oder Verhindern des automatischen Starts	52
Festlegen der Startparameter für das Datenbankmodul	54
Verwalten von Diensten an der Befehlszeile	56
Verwalten der ausführbaren SQL Server-Datei an der Befehlszeile	57
Verwalten der Serveraktivität	58
Auswerten von Prozessinformationen	58
Nachverfolgen von Ressourcenwartevorgängen und Blockierungen	59
Fehlerbehebung bei Deadlocks und blockierenden Verbindungen	62
Protokollierung der Befehlsausführung in SQL Server	64
Abbrechen von Serverprozessen	65
2 Verwalten von SQL Server-Diensten und -Clients	67
Verwalten des Zugriffs auf SQL Server-Komponenten	68
Verwalten der Netzwerk- und der SQL Native Client-Konfiguration	70
Verwalten der Verbindungskonfiguration	72
Festlegen der Shared-Memory-Netzwerkconfiguration	73
Festlegen der Named-Pipes-Netzwerkconfiguration	73
Festlegen der TCP/IP-Netzwerkconfiguration	74
Einrichten der Sicherheit für Native Client-Konfigurationen	78
Einrichten der Protokollreihenfolge	78
Einrichten von Shared Memory für eine Native Client-Konfiguration	79
Einrichten von Named Pipes für eine Native Client-Konfiguration	79
Einrichten von TCP/IP für eine Native Client-Konfiguration	80
Grundlagen der Dienstkfiguration	80
Einrichten von SQL Server-Diensten	84
Verwalten von Dienststatus und Startmodus	86
Festlegen des Dienststartkontos	87
Einrichten von FILESTREAM	89
Einrichten von Dienstspeicherverzeichnissen, Fehlerberichterstattung und der Berichterstellung für Kundenfeedback	91
Teil II	
Verwaltung und Sicherheit von SQL Server 2012	93
3 Richtliniengestützte Verwaltung	95
Einführung in die richtliniengestützte Verwaltung	96
Arbeiten mit der richtliniengestützten Verwaltung	99
Einrichten von zentralen Verwaltungsservern	104
Registrieren von zentralen Verwaltungsservern	104
Registrieren von untergeordneten Servern und Gruppen	105
Verschieben von untergeordneten Servern und Servergruppen	106
Löschen von untergeordneten Servern und Servergruppen	107
Ausführen von Anweisungen auf mehreren Servern	107
Unternehmensweite Richtlinienvverwaltung	108
Importieren und Exportieren von Richtlinien	108
Einrichten und Verwalten von Facets	109

Erstellen und Verwalten von Richtlinienbedingungen	112
Erstellen und Verwalten von Richtlinien	115
Verwalten von Richtlinienkategorien und Vorschreiben von Richtlinien	118
Auswerten von Richtlinien	120
Beheben von Richtlinienfehlern	122
4 Konfigurieren und Optimieren von SQL Server-Computern	125
Zugriff auf die Konfigurationsdaten von SQL Server	126
Vorgehensweisen zur Verwaltung der Konfigurationsoptionen von SQL Server	129
Festlegen von Konfigurationsoptionen	129
Arbeiten mit SET-Optionen	130
Arbeiten mit Serveroptionen	132
Arbeiten mit Datenbankoptionen	133
Verwalten der Datenbankkompatibilität	134
Konfigurieren von SQL Server mit gespeicherten Prozeduren	135
Verwenden von SQL Server Management Studio für Abfragen	136
Ausführen von Abfragen und Ändern von Einstellungen	137
Überprüfen und Festlegen von Konfigurationsparametern	139
Ändern von Einstellungen mit ALTER DATABASE	143
5 Anpassen und Verknüpfen von SQL Server-Computern	147
Grundlagen von SQL Server Management Studio	148
Verwalten der Konfiguration mit SQL Server Management Studio	148
Ermitteln von System- und Servereigenschaften	150
Einrichten von Hilfsprogramm-Steuerungspunkten	151
Arbeiten mit Verbindungsservern und verteilten Daten	157
Verteilte Abfragen	158
Verteilte Transaktionen	160
Ausführen des DTC-Dienstes	162
Einrichten von Authentifizierung und Überwachung	162
Festlegen des Authentifizierungsmodus	162
Festlegen der Überwachungsstufe	163
Aktivieren und Deaktivieren der C2-Überwachungsprotokollierung	164
Aktivieren und Deaktivieren der CC-Konformität	165
Optimieren der Speicherauslastung	166
Arbeiten mit dynamisch konfiguriertem Speicher	167
Verwenden von festem Speicher	168
Aktivieren der AWE-Speicherunterstützung	169
Optimieren des Speichers für die Indizierung	170
Zuweisen von Speicher für Abfragen	171
Konfigurieren von Prozessoren und Parallelverarbeitung	172
Optimieren der Prozessornutzung	172
Festlegen der Parallelverarbeitung	175
Einrichten von Threading, Prioritäten und Fibers	176
Einrichten von Benutzer- und Remoteverbindungen	178
Festlegen der Höchstzahl von Benutzerverbindungen	179
Festlegen der Standardverbindungsoptionen	179
Einrichten der Remoteserververbindungen	181

Verwalten von Servereinstellungen	182
Aktivieren und Deaktivieren der Unterstützung für eigenständige Datenbanken	183
Aktivieren und Deaktivieren der Unterstützung für FILESTREAM	183
Einstellen der Standardsprache für SQL Server	184
Zulassen und Verbieten von verschachtelten Triggern	185
Steuern der Abfrageausführung	185
Einrichten der Jahr-2000-Unterstützung	186
Verwalten von Datenbankeinstellungen	186
Festlegen des Indexfüllfaktors	187
Festlegen der Timeoutoptionen für Sicherung und Wiederherstellung	188
Festlegen der Beibehaltungszeit für Sicherung und Wiederherstellung	189
Entleeren des Caches an Prüfpunkten	189
Komprimieren der Sicherung	190
Verwalten von Verbindungsservern	190
Hinzufügen von Verbindungsservern	190
Einrichten der Sicherheit für Verbindungsserver	194
Festlegen von Optionen für Remote- und Verbindungsserver	196
Löschen von Verbindungsservern	197
Beheben von Konfigurationsproblemen	198
Reparieren einer fehlerhaften Konfiguration	198
Ändern der Sortierung und Wiederherstellen der master-Datenbank	199
6 Wesentliche Aufgaben der Datenbankverwaltung	201
Datenbankdateien und Protokolle	202
Allgemeine Datenbankverwaltung	206
Anzeigen von Datenbankinformationen in SQL Server Management Studio	207
Anzeigen von Datenbankinformationen mit T-SQL	209
System- und Beispieldatenbanken	210
Untersuchen von Datenbankobjekten	210
Erstellen von Datenbanken	213
Erstellen von Datenbanken in SQL Server Management Studio	213
Erstellen von Datenbanken mit T-SQL	217
Ändern von Datenbanken und ihren Optionen	218
Festlegen von Datenbankoptionen in SQL Server Management Studio	219
Ändern von Datenbanken mit ALTER DATABASE	220
Einrichten von automatischen Optionen	225
Steuern der ANSI-Konformität für einzelne Datenbanken	227
Einrichten der Parametrisierung	229
Einrichten von Cursoroptionen	230
Steuern des Benutzerzugriffs und des Datenbankstatus	232
Einstellen des Online-, Offline- und Notfallmodus	233
Optionen für die datenbankübergreifende Besitzverketzung und den externen Zugriff	234
Wiederherstellung, Protokollierung und Fehlerprüfung für die Festplatten-E/A	235
Anzeigen, Ändern und Überschreiben von Datenbankoptionen	236
Verwalten der Datenbank- und Protokollgröße	237
Automatische Verwaltung der Dateigröße	237
Manuelles Erweitern von Datenbanken und Protokollen	238
Manuelles Komprimieren und Verkleinern einer Datenbank	238

Bearbeiten von Datenbanken	242
Umbenennen von Datenbanken	242
Löschen von Datenbanken	243
Anfügen und Trennen von Datenbanken	244
Tipps und Techniken	248
Kopieren und Verschieben von Datenbanken	248
Verschieben von Datenbanken	252
Verschieben von tempdb und Ändern der Größe	253
Erstellen von sekundären Daten- und Protokolldateien	254
Verhindern von Fehlern des Transaktionsprotokolls	255
Verhindern von vollen Dateigruppen	255
Erstellen einer neuen Datenbankvorlage	256
Einrichten der Datenbankverschlüsselung	256
7 Sicherheit in SQL Server 2012	259
Überblick über die Sicherheit in SQL Server 2012	260
Arbeiten mit Sicherheitsprinzipalen und sicherungsfähigen Elementen	261
Grundlagen der Berechtigungen für sicherungsfähige Elemente	263
Untersuchen der Berechtigungen für sicherungsfähige Elemente	264
Authentifizierungsmodi in SQL Server 2012	267
Windows-Authentifizierung	268
Gemischte Sicherheit und SQL Server-Anmeldungen	268
Authentifizierung auf Datenbankebene	269
Anmeldungen und Benutzer für besondere Zwecke	269
Die Gruppe Administratoren	270
Das Konto Administrator	270
Die Anmeldung sa	270
Die Anmeldungen NT SERVICE und SYSTEM	271
Der Benutzer guest	271
Der Benutzer dbo	271
Die Benutzer sys und INFORMATION_SCHEMA	272
Berechtigungen	272
Objektberechtigungen	272
Anweisungsberechtigungen	275
Implizite Berechtigungen	276
Rollen	276
Serverrollen	276
Datenbankrollen	278
Verwalten von Serveranmeldungen	280
Anzeigen und Bearbeiten bestehender Anmeldungen	280
Erstellen von Anmeldungen	282
Gewähren und Verweigern des Serverzugriffs	285
Aktivieren, Deaktivieren und Entsperren von Anmeldungen	286
Entfernen von Anmeldungen	287
Ändern von Kennwörtern	288
Einrichten von Serverrollen	288
Zuweisen von Rollen zu Anmeldungen	288
Zuweisen von Rollen zu mehreren Anmeldungen	290
Aufheben von Zugriffsrechten und Rollen für Anmeldungen	291

Steuern des Datenbankzugriffs und der Verwaltung	291
Anzeigen und Bearbeiten vorhandener Datenbankbenutzer	291
Erstellen von Datenbankbenutzern	292
Benutzer mit T-SQL bearbeiten	294
Benutzer entfernen	295
Zuweisen von Zugriffsberechtigungen und Rollen zu einer Anmeldung	296
Zuweisen von Rollen für mehrere Anmeldungen	297
Erstellen von Standarddatenbankrollen	298
Erstellen von Anwendungsrollen	299
Entfernen der Rollenmitgliedschaften von Datenbankbenutzern	300
Löschen benutzerdefinierter Rollen	301
T-SQL-Befehle für die Zugriffs- und Rollenverwaltung	301
Verwalten von Datenbankberechtigungen	302
Zuweisen von Datenbankberechtigungen für Anweisungen	303
Objektberechtigungen für einzelne Anmeldungen	307
Objektberechtigungen für mehrere Anmeldungen	309
Teil III	
Datenverwaltung mit SQL Server 2012	311
8 Bearbeiten von Schemas, Tabellen und Sichten	313
Arbeiten mit Schemas	314
Erstellen von Schemas	315
Ändern von Schemas	317
Verschieben von Objekten zu einem neuen Schema	318
Entfernen von Schemas	319
Erste Schritte mit Tabellen	320
Grundlagen von Tabellen	321
Grundlagen von Datenseiten	321
Grundlagen von Blöcken	323
Grundlagen von Tabellenpartitionen	323
Arbeiten mit Tabellen	324
Erstellen von Tabellen	325
Ändern bestehender Tabellen	330
Anzeigen von Informationen über Tabellenzeilen und Größe	333
Anzeigen von Tabelleneigenschaften und Berechtigungen	333
Anzeigen der aktuellen Werte in einer Tabelle	334
Kopieren von Tabellen	334
Umbenennen und Löschen von Tabellen	335
Hinzufügen und Entfernen von Tabellenspalten	336
Skriptgesteuertes Erstellen von Tabellen	336
Verwalten von Tabellenwerten	337
Verwenden von systemeigenen Datentypen	337
Felder mit fester, variabler und maximaler Länge	341
Benutzerdefinierte Datentypen	341

Zulassen und Verweigern von NULL-Werten	344
Standardwerte	344
Spalten mit geringer Dichte	345
Identitäten und global eindeutige Bezeichner	346
Benutzerdefinierte Tabellentypen	347
Verwenden von Sichten	351
Arbeiten mit Sichten	351
Erstellen von Sichten	352
Ändern von Sichten	356
Aktualisierbare Sichten	356
Verwalten von Sichten	357
9 Indizes, Einschränkungen und Partitionen	359
Erstellen und Verwalten von Indizes	360
Grundlagen von herkömmlichen Indizes	360
Gruppierte Indizes	363
Nicht gruppierte Indizes	363
XML-Indizes	363
Gefilterte Indizes	364
Welche Spalten müssen indiziert werden?	365
Indizieren von berechneten Spalten und Sichten	366
Anzeigen von Indizeigenschaften	366
Erstellen von Indizes	368
Verwalten von Indizes	372
Verwenden des Datenbankoptimierungsratgebers	375
Arbeiten mit Columnstore-Indizes	379
Verwenden von Columnstore-Indizes	380
Anzeigen der Eigenschaften von Columnstore-Indizes	381
Erstellen und Verwalten von Columnstore-Indizes	382
Einschränkungen und Regeln für Spalten	383
Einschränkungen	384
Regeln	388
Erstellen von partitionierten Tabellen und Indizes	389
Erstellen von Partitionsfunktionen	389
Erstellen von Partitionsschemas	390
Erstellen von Partitionen	391
Anzeigen und Verwalten von Partitionen	392
Komprimieren von Tabellen, Indizes und Partitionen	394
Zeilen- und Seitenkomprimierung	394
Festlegen und Ändern von Komprimierungseinstellungen	396

Teil IV

Optimierung, Wartung und Wiederherstellung von SQL Server 2012

	397
10 Automatisierung und Wartung von SQL Server 2012	399
Datenbankautomatisierung und -wartung	400
Datenbank-E-Mail	402
Erstkonfiguration von Datenbank-E-Mail	402
Verwalten von Profilen und Konten für Datenbank-E-Mail	406
Einsehen und Ändern von Systemparametern für Datenbank-E-Mail	408
Verwenden des SQL Server-Agents	408
Zugriff auf Warnungen, Operatoren und Aufträge	408
Einrichten des SQL Server-Agent-Dienstes	409
Einrichten des E-Mail-Profiles für den SQL Server-Agent	409
Automatischer Neustart von Diensten mithilfe des SQL Server-Agents	410
Verwalten von Warnungen	410
Standardwarnungen	410
Erstellen von Warnungen für Fehlermeldungen	411
Reagieren auf Warnungen	412
Löschen, Aktivieren und Deaktivieren von Warnungen	414
Verwalten von Operatoren	414
Registrieren von Operatoren	414
Löschen und Deaktivieren der Benachrichtigungen für Operatoren	416
Einrichten eines Ausfallsicherheitsoperators	416
Planen von Aufträgen	417
Erstellen von Aufträgen	417
Zuweisen und Ändern von Auftragsdefinitionen	417
Festlegen der auszuführenden Schritte	419
Einrichten der Zeitpläne für Aufträge	423
Handhaben von Auftragswarnungen	425
Handhaben von Benachrichtigungen	426
Verwalten bestehender Aufträge	427
Verwalten von Auftragskategorien	427
Automatisieren von Routineaufgaben der serverübergreifenden Verwaltung	428
Kopieren von Benutzern, Tabellen, Sichten und anderen Objekten von einer Datenbank zu einer anderen	429
Kopieren von Warnungen, Operatoren und geplanten Aufträgen von einem Server auf einen anderen	431
Verwaltung mehrerer Server	432
Ereignisweiterleitung	432
Serverübergreifende Auftragsverwaltung	433
Datenbankwartung	435
Checklisten für die Datenbankwartung	435
Verwenden von Wartungsplänen	436
Überprüfen und Erhalten der Datenbankintegrität	441

11	Sichern und Wiederherstellen von SQL Server 2012	447
	Aufstellen eines Sicherungs- und Wiederherstellungsplans	448
	Erste Schritte bei der Planung	448
	Spiegelung und Sicherung gespiegelter Datenbanken	453
	Sicherung replizierter Datenbanken	453
	Sicherung sehr großer Datenbanken	455
	Komprimierung der Sicherungen	455
	Auswählen von Sicherungsgeräten und -medien	456
	Sicherungsverfahren	459
	Erstellen eines Sicherungsmediums	461
	Durchführen von Sicherungen	463
	Durchführen von Sicherungen in SQL Server Management Studio	463
	Stripesetsicherungen auf mehreren Geräten	467
	Verwenden der T-SQL-Anweisung BACKUP	468
	Sichern der Transaktionsprotokolle	472
	Wiederherstellen einer Datenbank	474
	Fehlerbehebung bei beschädigten Datenbanken	474
	Wiederherstellen einer Datenbank aus einer normalen Sicherung	476
	Wiederherstellen von Dateien und Dateigruppen	482
	Wiederherstellen einer Datenbank an einem anderen Speicherort	483
	Wiederherstellen fehlender Daten	484
	Erstellen von Standbyservern	485
	Verwenden von T-SQL-Befehlen zur Wiederherstellung	486
	Wiederherstellen der master-Datenbank	492
12	Profilerstellung und Überwachung von SQL Server 2012	493
	Überwachen der Serverleistung und -aktivität	494
	Gründe für die Überwachung von SQL Server	494
	Vorbereitung der Überwachung	494
	Werkzeuge und Informationsquellen für die Überwachung	495
	Arbeiten mit dem Replikationsmonitor	497
	Starten und Verwenden des Replikationsmonitors	498
	Hinzufügen von Verlegern und Verlegergruppen	499
	Arbeiten mit Ereignisprotokollen	500
	Untersuchen des Anwendungsprotokolls	501
	Untersuchen der SQL Server-Ereignisprotokolle	503
	Untersuchen der Ereignisprotokolle des SQL Server-Agents	505
	Überwachen der Leistung von SQL Server	506
	Auswählen der zu überwachenden Leistungsindikatoren	506
	Leistungsprotokollierung	509
	Anzeigen von Berichten	513
	Einrichten von Leistungsindikatorenwarnungen	514
	Einrichten eines Verwaltungs-Data Warehouse	515
	Grundlagen des Verwaltungs-Data Warehouses	515
	Erstellen eines Verwaltungs-Data Warehouses	515
	Einrichten einer Datenauflistung	516
	Verwalten von Datenauflistungen und Erstellen von Berichten	516

Beheben von Leistungsproblemen mit Profiler	517
Verwenden von Profiler	517
Erstellen von neuen Ablaufverfolgungen	518
Arbeiten mit Ablaufverfolgungen	522
Speichern von Ablaufverfolgungen	522
Wiedergeben von Ablaufverfolgungen	523
Stichwortverzeichnis	527
Autor und Fachgutachter	543

Für meine Frau – viele Jahre lang, über viele Bücher, Millionen von Worten und vielen Tausenden von Seiten hinweg war sie war mich da, gab mir Unterstützung und Ermutigung und machte jeden Ort, an dem wir lebten, zu einem Zuhause.

Für meine Kinder – Dafür, dass sie mir helfen, die Welt auf andere Weise zu sehen, für ihre außerordentliche Geduld und grenzenlose Liebe, und dafür, dass sie jeden Tag zu einem Abenteuer machen.

Für Karen, Martin, Lucinda, Juliana, Ben und alle anderen, die mir in großen und in kleinen Dingen geholfen haben.

– William R. Stanek

Einleitung

In dieser Einleitung:

Für wen ist dieser Ratgeber konzipiert?	20
Wie ist dieses Buch aufgebaut?	21
Was ist SQL Server 2012?	21
Systemvoraussetzungen	22
In diesem Buch verwendete Konventionen	24
Weitere Quellen	24
Support	25

Microsoft SQL Server 2012 – Ratgeber für Administratoren wurde als umfassende und besonders nützliche Ressource für SQL Server 2012-Administratoren konzipiert. Dieses Buch enthält alles, was Sie wissen müssen, um grundlegende Verwaltungsaufgaben in SQL Server durchführen zu können. Diese gut lesbare technische Referenz werden Sie immer in Reichweite haben wollen. Da Ihnen hier umfangreiche Informationen in Form eines Taschenbuchs vorliegen, müssen Sie sich nicht durch Hunderte von Seiten irrelevanter Angaben kämpfen, um die benötigten Hinweise zu finden. Sie erhalten genau die Informationen, die Sie zur Durchführung einer Aufgabe benötigen.

Dieses Buch soll Ihre zentrale Quelle zur Beantwortung aller Fragen rund um die Verwaltung von SQL Server sein. Darum liegt das Hauptaugenmerk auf den täglichen Verwaltungsverfahren, auf häufigen Aufgaben, gut dokumentierten Beispielen und repräsentativen Optionen, aber ohne Anspruch auf Vollständigkeit. Eines der Hauptziele bestand darin, den Inhalt so kurz und präzise wie möglich zu halten, damit das Buch einerseits kompakt und leicht zu handhaben ist, andererseits aber so viele Informationen enthält wie möglich. Statt eines dicken 1000-seitigen Wälzers oder einer dünnen 100-seitigen Kurzreferenz erhalten Sie eine nützliche Anleitung, die Ihnen bei allgemeinen Aufgaben, beim Lösen von Problemen und beim Einrichten erweiterter SQL Server-Technologien wie Replikation, verteilte Abfragen und Verwaltung mehrerer Server schnelle Hilfe bietet.

Für wen ist dieser Ratgeber konzipiert?

Microsoft SQL Server 2012 – Ratgeber für Administratoren behandelt die Standard, Business Intelligence, Enterprise und Developer Edition von SQL Server. Dieses Buch wurde für die Verwendung bei der täglichen Verwaltung von SQL Server konzipiert und für folgende Personengruppen geschrieben:

- SQL Server-Datenbankadministratoren
- Fortgeschrittene Benutzer, die auch einige Administratortasken übernehmen
- Administratoren, die von einer vorherigen Version zu SQL Server 2012 wechseln
- Administratoren, die von einer anderen Datenbankarchitektur wechseln

Um so viele Informationen wie möglich zur Verfügung zu stellen, bin ich davon ausgegangen, dass Sie grundlegende Fähigkeiten im Umgang mit Netzwerken haben und über ein grundlegendes Verständnis von SQL Server verfügen. In Anbetracht dessen habe ich der SQL Server-Architektur und einfachen SQL-Abfragen keine eigenen Kapitel gewidmet. Enthalten sind jedoch die Installation und Konfiguration von SQL Server, die unternehmensweite Serververwaltung, Leistungssteigerung, Optimierung, Wartung und viele andere Themen.

Des Weiteren bin ich davon ausgegangen, dass Sie mit den SQL-Befehlen und den gespeicherten Prozeduren genauso vertraut sind wie mit der Benutzeroberfläche von Windows. Wenn Sie die SQL-Grundlagen kennen lernen wollen, stehen Ihnen dazu viele andere Bücher zur Verfügung (darunter auch sehr viele von Microsoft Press).

Wie ist dieses Buch aufgebaut?

Die schnelle und einfache Handhabung ist wesentliches Merkmal dieses Buchs. Es verfügt über ein erweitertes Inhaltsverzeichnis und einen umfassenden Index, um schnell Lösungsvorschläge für Problemfälle zu finden. Ebenso wurde eine große Anzahl von weiteren Nachschlagemöglichkeiten hinzugefügt, um Probleme schnell zu lösen. Dazu gehören Schritt-für-Schritt-Anleitungen, Listen, Tabellen mit übersichtlichen Informationen und Querverweise.

Der Inhalt ist in vier Teile gegliedert:

- Teil I, »Grundlagen von Microsoft SQL Server 2012«, beschreibt den Umgang mit SQL Server und mit SQL Server-Diensten und Clients.
- In Teil II, »Verwaltung und Sicherheit von SQL Server 2012«, geht es um die Implementierung und Konfiguration der SQL Server-Umgebung.
- Teil III, »Datenverwaltung mit SQL Server 2012«, legt den Schwerpunkt auf Daten, auf tägliche Aufgaben und auf empfohlene Vorgehensweisen für die Verwaltung der Daten.
- Teil IV, »Optimierung, Wartung und Wiederherstellung von SQL Server 2012«, behandelt einige der anspruchsvolleren Themen, mit denen alle Administratoren vertraut sein müssen.

Was ist SQL Server 2012?

SQL Server 2012 fungiert als Plattform für geschäftsentscheidende Daten, ermöglicht eine dynamische Entwicklung, bietet ausführliche Möglichkeiten für Business Intelligence und geht weit über relationale Daten hinaus. Damit bietet SQL Server 2012 eine solide Grundlage, auf der kleine, mittlere und große Organisationen ihre IT-Infrastruktur aufbauen können. Die Kernbestandteile von SQL Server 2012 sind:

- **Database Engine Services** Endhält die Kernkomponenten für Datenbanken, Benachrichtigungen und Replikation. Die Kerndatenbank – das Datenbankmodul – ist das Herz von SQL Server. Replikation verteilt die Daten auf mehrere Datenbanken, was ihre Verfügbarkeit erhöht und Ihnen die Möglichkeit gibt, die Lesebelastung auf dafür vorgesehene Datenbankserver auszulagern.
- **Analysis Services** Ermöglicht OLAP- (Online Analytical Processing) und Data-Mining-Funktionen für Business-Intelligence-Anwendungen. Analysis Services versetzt Ihre Organisation in die Lage, Daten aus mehreren Quellen zusammenzufassen, z.B. aus mehreren relationalen Datenbanken, um auf verschiedene Weise mit ihnen zu arbeiten.
- **Integration Services** Bietet eine professionelle Datentransformations- und Integrationslösung, um Daten aus mehreren Datenquellen zu entnehmen, zu transformieren und in eine oder mehrere Zieldatenquellen zu verschieben. Damit können Sie Daten aus heterogenen Datenquellen verschmelzen, in Data Warehouses und Data Marts laden usw.

- **Reporting Services** Umfasst Report Manager und Report Server, die eine vollständige servergestützte Plattform zum Erstellen, Verwalten und Verteilen von Berichten bilden. Report Server baut auf den Standardtechnologien der Internetinformationsdienste (IIS) und von Microsoft .NET Framework auf und ermöglicht es Ihnen, die Vorteile von SQL Server und IIS zu kombinieren und Berichte bereitzustellen und zu verarbeiten.
- **Service Broker** Bietet zuverlässige Warteschlangen- und Nachrichten als zentraler Bestandteil der Datenbank. Warteschlangen können dazu dienen, Aufgaben wie Abfragen und andere Anforderungen zu sammeln und so durchzuführen, wie es die Ressourcen erlauben. Durch Nachrichten können Datenbankanwendungen miteinander kommunizieren. Das Datenbankmodul setzt Service Broker ein, um Benachrichtigungen zu verschicken.
- **Master Data Services** Bietet ein Framework, um Geschäftsregeln zu erstellen, mit denen die Qualität und Genauigkeit der Masterdaten sichergestellt wird. Geschäftsregeln können für Geschäftsvorgänge eingesetzt werden, die Validierungsprobleme korrigieren und Arbeitsabläufe handhaben.
- **Data Quality Services** Bietet ein Framework, um eine Ablage von Metadaten zu erstellen und damit die Qualität der Organisationsdaten zu verbessern. Verfahren zur Datenbereinigung können unvollständige oder fehlerhafte Daten entfernen. Mit Datenvergleichsprozessen können Duplikate erkannt und nach Bedarf zusammengeführt werden.

Systemvoraussetzungen

Für die erfolgreiche Verwaltung von Datenbankservern sind drei Dinge erforderlich: sachkundige Datenbankadministratoren, eine solide Datenbankarchitektur und geeignete Hardware. Die ersten beiden Zutaten können Sie schon abhaken: Sie sind der Administrator, Sie waren klug genug, sich dieses Buch zu kaufen, um die gefährlichen Klippen zu umschiffen, und Sie haben SQL Server 2012 eingerichtet, um Ihren Bedarf nach hochleistungsfähigen Datenbanken zu befriedigen. Bleibt also noch die Frage der Hardware. Führen Sie SQL Server 2012 auf einem System mit ausreichend Arbeitsspeicher, geeigneter Verarbeitungsgeschwindigkeit und genügend Festplattenplatz aus. Außerdem muss die Hardware die Bedürfnisse an den Daten- und Systemschutz erfüllen.

Befolgen Sie bei der Auswahl von Hardware für SQL Server die folgenden Richtlinien:

- **Arbeitsspeicher** Für alle Editionen von SQL Server 2012 außer Express ist mindestens 1 GB RAM erforderlich. Als Ausgangspunkt sollten Sie jedoch in den meisten Fällen über mindestens 4 GB RAM verfügen, selbst für Entwicklungsaufgaben. Der Hauptgrund für dieses Mehr an Arbeitsspeicher ist die Leistung. Zusätzliche Datenbankfunktionen – wie Analysis Services, Reporting Services und Integration Services – erhöhen die Anforderungen an den Arbeitsspeicher. Berücksichtigen Sie auch die Anzahl der Benutzerverbindungen, die jeweils etwa 24 KB in Anspruch nehmen. Auch Datenanforderungen und andere SQL Server-Prozesse nutzen Arbeitsspeicher, was noch zum Speicherverbrauch durch alle anderen Prozesse und Anwendungen auf dem Server hinzukommt.
- **Prozessor** Die 64-Bit-Versionen laufen auf den x64-Prozessoren von AMD und Intel, z.B. auf dem AMD64 und dem Intel EM64T (Extended Memory 64 Technology). Empfohlen werden die Mehrkernprozessoren Intel Xeon und AMD Opteron. SQL Server 2012 unterstützt symmetrisches Multiprocessing

und kann komplexe parallele Abfragen verarbeiten. Das ist jedoch nur dann von Vorteil, wenn relativ wenige Benutzer das System verwenden und die zu verarbeitenden Abfragen umfangreich sind. Auf einem System, auf dem ausschließlich SQL Server ausgeführt wird, auf das weniger als 100 Benutzer gleichzeitig zugreifen und auf dem keine komplizierten Abfragen verarbeitet werden, reicht ein einzelner Mehrkernprozessor aus. (Allerdings sollten Sie dies immer mit einer repräsentativen Arbeitslast testen.) Bei mehr als 100 Benutzern oder einem System, das auch noch andere Aufgaben erfüllen muss, sollten Sie weitere Prozessoren hinzufügen (oder ein System verwenden, das Sie bei wachsendem Bedarf um weitere Prozessoren ergänzen können). Denken Sie auch daran, dass sich der Umfang der zu verarbeitenden Abfragen und Datasets darauf auswirkt, wie gut sich SQL Server wachsenden Bedingungen anpassen kann. Je größer die Aufträge werden, die SQL Server verarbeiten muss, umso mehr Arbeitsspeicher und Prozessorkapazität brauchen Sie.

- **Festplattenlaufwerke** Wie groß die Kapazität zur Speicherung von Daten sein muss, hängt von der Anzahl und Größe der Datenbanken auf dem Server ab. Sie brauchen genügend Platz, um alle Daten und darüber hinaus die Arbeitsbereiche, Indizes, Systemdateien, Transaktionsprotokolle und den virtuellen Speicher unterzubringen. Für Protokollversand und Spiegelung brauchen Sie Platz für die Sicherungsfreigabe, und bei einem Cluster ist auch eine Quorumfestplatte erforderlich. Der E/A-Durchsatz ist genauso wichtig wie das Fassungsvermögen der Festplatten. Um die bestmögliche E/A-Leistung zu erzielen, werden Fibre-Channel- oder FCoE-Lösungen (Fibre Channel over Ethernet) als anspruchsvolle Speicherlösungen empfohlen. Sie sollten auch lieber SSD-Laufwerke (Solid State Drives) anstelle von rotierenden Plattenlaufwerken verwenden. Wenn Sie mehrere kleinere statt einer großen Festplatte einsetzen, können Sie mit RAID für Fehlertoleranz sorgen. Ich rate Ihnen dazu, Daten und Protokolle auf getrennten Platten unterzubringen. Das gilt auch für die Sicherungsfreigabe beim Protokollversand und die Quorumfestplatte eines Clusters.
- **Datensicherheit** Wappnen Sie sich mit RAID gegen unvorhersehbare Laufwerksausfälle. Beispielsweise können Sie RAID 0+1 oder RAID 5 verwenden oder für Protokolle RAID 1. RAID 0 (Stripeset ohne Parität) bietet eine gute Lese- und Schreibleistung, aber wenn irgendein Laufwerk ausfällt, kann SQL Server nicht mehr mit der betroffenen Datenbank arbeiten, bis das Laufwerk ersetzt und die Daten von einer Sicherung wiederhergestellt worden sind. RAID 1 (Festplattenspiegelung) erstellt eine Kopie der Daten auf einem zweiten Laufwerk. Eine solche RAID-Einheit können Sie zur vollen Betriebsbereitschaft wiederherstellen. RAID 5 (Stripeset mit Parität) bietet einen guten Schutz gegen den Ausfall eines einzelnen Laufwerks, aber eine schlechte Schreibleistung. Für eine gute Leistung und Fehlertoleranz wird RAID 0+1 empfohlen. Dabei werden Spiegelung und Stripeset ohne Parität kombiniert.
- **Unterbrechungsfreie Stromversorgung (USV)** SQL Server ist dazu entworfen, die Integrität der Datenbanken zu bewahren, und kann Informationen mithilfe von Transaktionsprotokollen wiederherstellen. Das schützt jedoch nicht die Hardware gegen plötzliche Stromausfälle oder Spannungsspitzen. Beides kann die Hardware ernsthaft beschädigen. Um das zu verhindern, sollten Sie sich eine USV zulegen, die die Stromzufuhr regelt. Ein solches Gerät gibt Ihnen Zeit, das System bei einem Stromausfall sauber herunterzufahren. Außerdem trägt es entscheidend zur Bewahrung der Datenbankintegrität bei, wenn der Server Write-Back-Cachecontroller verwendet.

Wenn Sie diese Richtlinien zur Hardware befolgen, haben Sie schon die ersten Schritte auf dem Weg zum Erfolg mit SQL Server 2012 gemacht.

In diesem Buch verwendete Konventionen

Verschiedene Elemente wurden verwendet, um dieses Buch klar und leicht verständlich zu gestalten. Listings werden in nichtproportionaler Schrift dargestellt. Wenn ein neuer Begriff eingeführt und definiert wird, erscheint er in *Kursivschrift*. Auch Dateinamen, Pfade und GUI-Begriffe werden so dargestellt.

Des Weiteren gibt es folgende Arten von Hinweiskästen:

Empfohlene Vorgehensweise

Erklärt die bestmögliche Vorgehensweise bei erweiterten Konfigurations- und Administrationsverfahren.

VORSICHT

Warnt vor eventuell auftretenden Problemen.

Weitere Informationen

Stellt mehr Informationen über ein Thema zur Verfügung.

HINWEIS

Gibt Einzelheiten zu einem Punkt an, der weiterer Erklärung bedarf.

Praxistipp

Bietet bei fortgeschrittenen Themen Tipps für die Praxis.

TIPP

Bietet hilfreiche Hinweise oder zusätzliche Informationen.

Ich hoffe, dass *Microsoft SQL Server 2012 – Ratgeber für Administratoren* Ihnen alle Informationen bietet, um wesentliche Verwaltungsaufgaben in SQL Server so schnell und effizient wie möglich erledigen zu können, und freue mich auf Ihre Kommentare (in englischer Sprache) unter williamstanek@aol.com. Besuchen Sie auch <http://www.williamstanek.com>. Vielen Dank.

Weitere Quellen

Es gibt ein Allzweckbuch, in dem Sie alles lernen könnten, was Sie jemals über SQL Server 2012 wissen müssen. Zwar werden einige Werke als Kompletthanleitung angepriesen, allerdings ist es schlicht nicht möglich, dass ein Buch alles abdeckt. Daher hoffe ich, dass Sie dieses Buch so einsetzen, wie es gedacht ist – als knappes und leicht zu verwendendes Nachschlagewerk. Es deckt alles ab, was Sie brauchen, um die wichtigsten Verwaltungsaufgaben für SQL Server zu erledigen, aber es erhebt auf keinen Fall Anspruch auf Vollständigkeit.

Wie viel Erfolg Sie mit diesem Buch oder irgendeiner anderen Informationsquelle zu SQL Server haben, hängt größtenteils von ihrem bisherigen Wissensstand ab. Wenn Sie auf Themen stoßen, die neu für Sie sind, nehmen Sie sich Zeit, das Gelernte zu üben. Forschen Sie nach zusätzlichen Informationen, um das Know-how und die praktischen Kenntnisse zu erwerben, die Sie benötigen.

Ich möchte Ihnen ans Herz legen, regelmäßig die Website zu SQL Server (www.microsoft.com/sqlserver/) und die Support-Website von Microsoft (www.support.microsoft.com) zu besuchen, um bei Änderungen immer auf dem neuesten Stand zu bleiben. Um den größtmöglichen Nutzen aus diesem Buch zu ziehen, besuchen Sie auch die Begleitwebsite unter www.williamstanek.com/sqlserver. Sie enthält Informationen über SQL Server 2012 und Aktualisierungen zu diesem Buch (in englischer Sprache).

Support

Wir haben uns um absolute Genauigkeit in diesem Buch bemüht. Microsoft Press bietet im World Wide Web Korrekturen und Support zu seinen Büchern unter der Adresse <http://www.microsoft-press.de/support.asp>.

Falls Sie Kommentare, Fragen oder Anregungen zu diesem Buch haben, senden Sie sie bitte an folgende Microsoft Press-Adresse:

mspressde@oreilly.de

Bitte beachten Sie, dass über die Mail-Adresse kein Softwareservice angeboten wird. Für Supportinformationen bezüglich der Softwareprodukte besuchen Sie die Microsoft-Website: <http://www.microsoft.com/germany/support>.

Teil I

Grundlagen von Microsoft SQL Server 2012

In diesem Teil:

Kapitel 1	Verwalten von SQL Server-Computern	29
Kapitel 2	Verwalten von SQL Server-Diensten und -Clients	67

Kapitel 1

Verwalten von SQL Server-Computern

In diesem Kapitel:

SQL Server Management Studio	31
Verwalten von SQL Server-Gruppen	35
Verwalten von Servern	38
Verwenden von Windows PowerShell zur Verwaltung von SQL Server	45
Starten, Beenden und Konfigurieren des SQL Server-Agents	51
Starten, Beenden und Konfigurieren von Microsoft Distributed Transaction Coordinator	51
Einrichten des Startverhaltens von SQL Server	52
Verwalten der Serveraktivität	58

SQL Server Management Studio ist das wichtigste Werkzeug zum Verwalten von Datenbanken und Servern. Für die Verwaltung von lokalen und Remoteservern stehen Ihnen darüber hinaus SQL Server PowerShell, der SQL Server-Konfigurations-Manager, die Leistungsüberwachung, der Datenbankoptimierungsratgeber und SQL Server Profiler zur Verfügung. Mit dem Konfigurations-Manager können Sie SQL Server-Dienste, Netzwerk- und Clientkonfigurationen verwalten. Der Datenbankoptimierungsratgeber dient zum Optimieren von Indizes, indizierten Sichten und Partitionen. Über SQL Server Profiler können Sie von SQL Server generierte Ereignisse untersuchen, die unter Umständen wichtige Hinweise zu Problemen enthalten. Dieses Kapitel beschäftigt sich mit SQL Server Management Studio. Der Konfigurations-Manager ist Thema von Kapitel 2, »Verwalten von SQL Server-Diensten und -Clients«. Informationen zur Optimierung und Ablaufverfolgung finden Sie in Kapitel 12, »Profilerstellung und Überwachung von SQL Server 2012«.

Denken Sie zur erfolgreichen Arbeit mit Datenbanken und Servern immer an folgende Prinzipien:

- **Eigenständige Datenbanken** Dabei handelt es sich um ganz oder teilweise isolierte Datenbanken, die auf der Instanz des SQL Server-Datenbankmoduls, auf der sie installiert sind, keinerlei Konfigurationsabhängigkeiten aufweisen. Eine vollständig eigenständige Datenbank lässt keine Objekte und Funktionen zu, die die Grenzen zwischen dem Anwendungsmodell und der Instanz des Datenbankmoduls überschreiten. Bei einer nur teilweise eigenständigen Datenbank sind solche Objekte und Funktionen jedoch erlaubt. Benutzer von eigenständigen Datenbanken mit Kennwörtern werden von der Datenbank authentifiziert. Autorisierte Windows-Benutzer und Gruppenmitglieder können direkt Verbindung mit der Datenbank aufnehmen und benötigen keine Anmeldung an der Master-Datenbank.
- **FileTable** Tabellenstrukturen fungieren als virtuelle Freigaben, indem sie FILESTREAM- und Verzeichnisdaten als Zeilen in Tabellen speichern. Das Datenbankmodul kümmert sich zwar zu jeder Zeit um die Verwaltung der Daten, doch für einen nicht auf Transaktionen gestützten Dateizugriff wird *FileTable* als Windows-Freigabe angezeigt. Dadurch können Sie MOVE, XCOPY und andere Standardbefehle zum Laden von Dateien einsetzen, wenn Sie an der Befehlszeile oder mit einem Batch-Skript arbeiten. Der Stamm der Hierarchie wird festgelegt, wenn Sie *FileTable* erstellen. *FileTable* kann nicht repliziert und auch nicht mit SELECT in andere Tabellen aufgenommen werden.
- **Indirekte Prüfpunkte** Prüfpunkte werden auf der Grundlage der angestrebten Wiederherstellungszeit ausgelöst, die Sie für eine Datenbank festgelegt haben. Im Gegensatz dazu wird bei automatischen Prüfpunkten die maximale Anzahl der Protokolleinträge herangezogen, die in einem bestimmten Wiederherstellungsintervall verarbeitet werden können. Eine Datenbank, für die eine Wiederherstellungszeit festgelegt ist, verwendet keine automatischen Prüfpunkte. Indirekte Prüfpunkte können das Auftreten von Spitzenwerten beim Lesen und Schreiben verringern, da alle Schreibaufgaben im Hintergrund erfolgen, doch wird dadurch die Gesamtbelastung der Serverinstanz für Schreibvorgänge erhöht. Das wiederum kann die Leistung für Onlinetransaktionen beeinträchtigen.

Berücksichtigen Sie auch die Änderungen an der Funktionsweise des Datenbankmoduls. Viele ältere Merkmale sind nicht mehr vorhanden. Bedenken Sie vor allem die folgenden wichtigen Änderungen:

- Für Datenbanken muss mindestens der Kompatibilitätsgrad 90 festgelegt sein, der für Microsoft SQL Server 2005 vorgesehen ist. Alle älteren Datenbanken werden automatisch aktualisiert, wenn Sie Microsoft SQL Server 2012 installieren.
- Indizes mit *varchar(max)*-, *nvarchar(max)*- und *varbinary(max)*-Spalten können jetzt online neu aufgebaut werden.

- Da Trigger mit WITH APPEND-Klauseln nicht mehr funktionieren, müssen Sie sie neu erstellen. Das gilt auch für COMPUTE- und COMPUTE BY-Klauseln, die durch ROLLUP-Klauseln ersetzt werden müssen.
- Ersetzen Sie Remoteserver durch Verbindungsserver und Aliasse durch Benutzerkonten und Datenbankrollen.
- Ersetzen Sie SQL Mail durch Datenbank-E-Mail und verwenden Sie ALTER DATABASE anstelle von *sp_dboption*.
- Verwenden Sie bei ALTER TABLE zweiteilige Tabellennamen nach dem Muster *schema.objekt* und keine vierteilige Namen wie *server.datenbank.schema.tabelle*.

SQL Server Management Studio

Sie können Datenbanken und Ressourcen über die grafische Benutzeroberfläche von SQL Server Management Studio Server im Handumdrehen verwalten. Dazu stellen Sie eine Verbindung zu SQL Server her und verwalten anschließend die lokalen und Remoteserverressourcen. Wenn Sie jedoch die Remoteserververbindungen deaktiviert haben, können Sie nur lokal arbeiten (indem Sie sich beim System direkt über die Tastatur anmelden oder indem Sie eine Remote-Terminalserversitzung unter Windows einrichten und anschließend die lokalen Verwaltungstools ausführen).

Erste Schritte mit SQL Server Management Studio

Wenn Sie mit SQL Server Management Studio arbeiten, sehen Sie als Erstes die Ansicht des Objektexplorers aus Abbildung 1.1 Wenn diese Ansicht nicht angezeigt wird, können Sie sie (und andere) über das Menü *Ansicht* öffnen. Die folgende Aufstellung erklärt, wie Sie die einzelnen Ansichten verwenden:

- **Objekt-Explorer** Hier können Sie die Instanzen von SQL Server, Analysis Services, Integration Services und Reporting Services einsehen und Verbindung zu ihnen aufnehmen. Nachdem Sie Verbindung zu einem bestimmten Server hergestellt haben, können Sie dessen Komponenten als Objektstruktur anzeigen lassen und die Knoten erweitern, um sich zu den tieferen Ebenen der Struktur vorzuarbeiten.
- **Registrierte Server** Zeigt die zurzeit registrierten Server an. Verwenden Sie Registrierte Server, um die Anmeldeinformationen für die Server festzuhalten, auf die Sie häufig zugreifen. In der oberen Leiste dieser Ansicht können Sie schnell zwischen den Servern eines bestimmten Typs umschalten (SQL Server, Analysis Server, Integration Server und Report Server).
- **Vorlagen-Explorer** Bietet einen schnellen Zugriff auf die (nach Aktionen geordneten) Standardvorlagen des Abfrage-Editors und auf von Ihnen selbst erstellte Vorlagen. Sie können Vorlagen in jeder von SQL Server Management Studio, SQL Server und Analysis Server unterstützten Sprache erstellen.
- **Lösungs-Explorer** Bietet einen schnellen Zugriff auf bestehende SQL Server- und Analysis Server-Projekte. Bei jedem Projekt sind die Verbindungen, Abfragen und sonstigen Funktionen angegeben, die beim Ausführen des Projekts ablaufen.

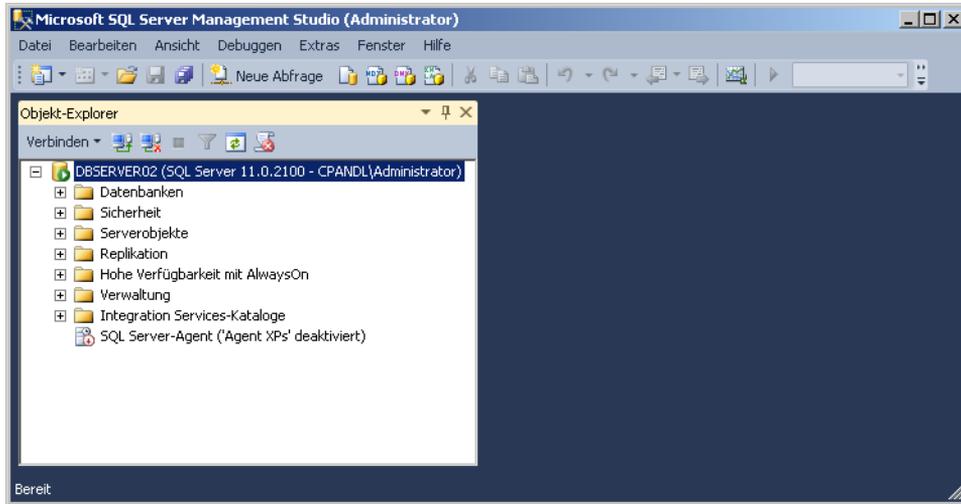


Abbildung 1.1 Zur Durchführung der wichtigsten Verwaltungsaufgaben verwenden Sie SQL Server Management Studio

Um SQL Server Management Studio auszuführen, klicken Sie auf *Start*, geben in das Suchfeld *ssms.exe* ein und drücken die Eingabetaste. Alternativ können Sie auch die entsprechende Option aus dem Menü *Microsoft SQL Server 2012* auswählen. Anschließend müssen Sie Verbindung mit dem Server aufnehmen, mit dem Sie arbeiten möchten. Dazu gibt es mehrere Möglichkeiten:

- Standardanmeldung an der Serverinstanz
- Anmeldung an einer bestimmten Datenbank
- Anmeldung über Servergruppen und registrierte Server

Wenn Sie Verbindung mit einer Serverinstanz aufnehmen, können Sie mit diesem Server und allen damit verbundenen Komponenten arbeiten. Gewöhnlich nehmen Sie dabei Verbindung mit dem Datenbankmodul des Servers auf (siehe Abbildung 5.2). Das Datenbankmodul gewährt Ihnen Zugriff auf folgende Komponenten:

- **Datenbanken** Sie können die Systemdatenbanken einschließlich *master* und *model* sowie Benutzerdatenbanken und Datenbank-Snapshots verwalten. Außerdem haben Sie unter diesem Knoten Zugriff auf die Datenbanken *ReportServer* und *ReportServerTmpDB*, sofern Sie Reporting Services installiert haben.
- **Sicherheit** Sie können SQL-Anmeldungen, Serverrollen, gespeicherte Anmeldeinformationen, Kryptographieanbieter und die Überwachung verwalten.
- **Serverobjekte** Hier können Sie Sicherungsgeräte, HTTP-Endpunkte, Verbindungsserver und Servertrigger konfigurieren.
- **Replikation** Sie können Verteilungsdatenbanken konfigurieren, die Replikationskennwörter aktualisieren und den Replikationsmonitor starten.
- **Verwaltung** Sie können SQL Server-Protokolle einsehen, Wartungspläne sowie Microsoft Distributed Transaction Coordinator (MSDTC) und Datenbank-E-Mail erstellen, anzeigen und verwalten, Datensammlungen, die Ressourcenkontrolle und die richtlinienbasierte Verwaltung konfigurieren.
- **SQL Server-Agent** Hier können Sie Aufträge des SQL Server-Agents, Warnungen, Operatoren, Proxys und Fehlerprotokolle einrichten.

Mit der Registrierungsfunktion speichern Sie Server- und Anmeldeinformationen. Registrierte Server können in Servergruppen gegliedert werden, woraufhin über die Ansicht *Registrierte Server* ein schneller Zugriff möglich ist. Wie Sie Servergruppen und registrierte Server verwalten, erfahren Sie in den Abschnitten »Verwalten von SQL Server-Gruppen« und »Verwalten von Servern«.



Abbildung 1.2 Verwenden des Datenbankmoduls zum Zugriff auf die Kernkomponenten und Funktionen von SQL Server

Verbindungen mit einer bestimmten Serverinstanz

Um sich mit einer Standardanmeldung mit einer bestimmten Serverinstanz zu verbinden, gehen Sie wie folgt vor:

1. Starten Sie SQL Server Management Studio. Wählen Sie im Dialogfeld *Verbindung mit Server herstellen* in der Dropdownliste *Servertyp* die Datenbankkomponente aus, zu der Sie eine Verbindung herstellen möchten, z.B. *Datenbankmodul*. (Falls Sie das Dialogfeld *Verbindung mit Server herstellen* ausgeblendet haben, können Sie es wieder anzeigen lassen, indem Sie in SQL Server Server Management Studio auf *Datei* und dann auf *Objekt-Explorer verbinden* klicken.)
2. Geben Sie im Feld *Servername* den vollständig qualifizierten Domännennamen (FQDN) oder den Hostnamen des Servers ein, auf dem SQL Server läuft, z.B. *EngDBSrv12.cpandl.com* oder *EngDBSrv12*. Sie können auch in der zugehörigen Dropdownliste *Suche fortsetzen* auswählen. Im Dialogfeld *Nach Servern suchen* wählen Sie die Registerkarte *Lokale Server* oder *Netzwerkserver*. Nach dem Abrufen der Instanzdaten erweitern Sie die angezeigten Knoten, wählen die gewünschte Serverinstanz und klicken auf OK.

HINWEIS

Die Liste im Dialogfeld *Nach Servern suchen* wird vom SQL Server-Browserdienst gefüllt, der auf Datenbankservern läuft. Es kann verschiedene Gründe dafür geben, dass eine SQL Server-Instanz, mit der Sie arbeiten möchten, nicht aufgeführt wird. Möglicherweise wird der SQL Server-Browserdienst auf dem SQL Server-Computer nicht ausgeführt. Es kann auch sein, dass eine Firewall den UDP-Port 1434 blockiert, der für den Dienst erforderlich ist, oder dass in der SQL Server-Instanz das Flag *HideInstance* gesetzt ist.

3. Wählen Sie aus der Dropdownliste *Authentifizierung* den Authentifizierungstyp, entweder Windows-Authentifizierung oder SQL Server-Authentifizierung (je nachdem, welche Authentifizierungstypen Sie bei der Installation des Servers zugelassen haben). Geben Sie einen Windows-Benutzernamen oder eine SQL Server-Anmelde-ID und ein Kennwort an.
 - **Windows-Authentifizierung** Hierbei werden Ihr Domänenkonto und Ihr Kennwort verwendet, um die Datenbankverbindung aufzubauen. Diese Art der Authentifizierung funktioniert nur, wenn die Windows-Authentifizierung aktiviert ist und Sie über die entsprechenden Rechte verfügen.
 - **SQL Server-Authentifizierung** Hierbei können Sie eine SQL Server-Anmelde-ID und ein Kennwort festlegen. Wählen Sie *Kennwort speichern*, um das Kennwort zu speichern, sodass Sie es nicht bei jedem Verbindungsaufbau neu eingeben müssen.
4. Klicken Sie auf *Verbinden*. Sie können jetzt die Objekt-Explorer-Ansicht verwenden, um mit dem Server zu arbeiten.

Verbindung mit einer bestimmten Datenbank

Um sich mit einer Standardanmeldung mit einer bestimmten Datenbank zu verbinden, gehen Sie wie folgt vor:

1. Starten Sie SQL Server Management Studio. Wählen Sie im Dialogfeld *Verbindung mit Server herstellen* aus der Dropdownliste *Servertyp* die Datenbankkomponente aus, zu der Sie eine Verbindung herstellen möchten, z.B. *Datenbankmodul*. Geben Sie dann im Feld *Servername* den vollständig qualifizierten Domännennamen oder den Hostnamen des Servers ein, auf dem SQL Server läuft, z.B. *EngDBSrv12.cpandl.com* oder *EngDBSrv12*. (Falls Sie das Dialogfeld *Verbindung mit Server herstellen* ausgeblendet haben, können Sie es wieder anzeigen lassen, indem Sie in SQL Server Server Management Studio auf *Datei* und dann auf *Objekt-Explorer verbinden* klicken.)
2. Wählen Sie aus der Dropdownliste *Authentifizierung* den Authentifizierungstyp, entweder Windows-Authentifizierung oder SQL Server-Authentifizierung (je nachdem, welche Authentifizierungstypen Sie bei der Installation des Servers zugelassen haben). Geben Sie ggf. eine SQL Server-Anmelde-ID und ein Kennwort an.
3. Klicken Sie auf *Optionen*, um die erweiterte Ansicht des Dialogfelds *Verbindung mit Server herstellen* ein-zublenden. Wählen Sie die Registerkarte *Verbindungseigenschaften*, die Sie in Abbildung 1.3 sehen.
4. Geben Sie den Namen der Datenbank ein, mit der Sie verbunden werden möchten, z.B. *Personnel*, oder wählen Sie *Server durchsuchen* von der zugehörigen Dropdownliste aus. Wenn Sie dazu aufgefordert werden, klicken Sie auf *Ja*, um die Verbindung zu einem zuvor angegebenen Server herzustellen. Wählen Sie im Dialogfeld *Server nach Datenbank durchsuchen* die Datenbank aus, die Sie verwenden möchten, und klicken Sie auf *OK*.
5. Wählen Sie aus der Liste *Netzwerkprotokolle* die Netzwerkprotokolle und sonstigen Verbindungseigen-schaften aus, wenn Sie dazu aufgefordert werden. Shared Memory ist das standardmäßige Netzwerkproto-koll für lokale Verbindungen, TCP/IP der Standard für Remoteverbindungen. Wenn Sie möchten, können Sie eine sichere Verbindung einrichten, indem Sie *Verbindung verschlüsseln* auswählen.
6. Klicken Sie auf *Verbinden*. Sie können jetzt in der Objekt-Explorer-Ansicht mit der gewünschten Daten-bank arbeiten.



Abbildung 1.3 Herstellen einer Verbindung zu einer bestimmten Datenbank

Verwalten von SQL Server-Gruppen

SQL Server-Gruppen dienen zum Ordnen von SQL Servern. Sie definieren diese Servergruppen und können sie unter anderem nach Funktion und Abteilung gliedern. Eine Servergruppe zu erstellen ist ein einfacher Vorgang. Sie können sogar Untergruppen innerhalb einer Gruppe anlegen. Wenn Ihnen ein Fehler unterläuft, können Sie die Gruppen auch löschen.

Weitere Informationen

Zentral verwaltete Server lassen sich ebenfalls in Servergruppen gliedern. Weitere Informationen erhalten Sie im Abschnitt »Einrichten von zentralen Verwaltungsservern« von Kapitel 3, »Richtliniengestützte Verwaltung«.

Einführung in SQL Server-Gruppen und die Ansicht Registrierte Server

In SQL Server Management Studio verwenden Sie für den Umgang mit Servergruppen die Ansicht *Registrierte Server*. Wenn Sie diese Ansicht nutzen wollen, drücken Sie **Strg** + **Alt** + **G**. Dadurch wird sie auch eingeblendet, wenn sie zuvor verborgen war.

Die Gruppen der obersten Ebene wurden auf der Grundlage der SQL Server-Instanzen bereits für Sie erstellt. Mit der Symbolleiste *Registrierte Server* können Sie zwischen den einzelnen Gruppen hin- und herschalten. Die Gruppen der obersten Ebene sind nach den SQL Server-Instanzen gegliedert:

- Datenbankmodul
- Analysis Services
- Reporting Services
- Datenbanken von SQL Server Mobile Edition
- Integration Services

Sie können registrierte Server zwar direkt zu den Gruppen der obersten Ebene hinzufügen (wie Sie im Abschnitt »Verwalten von Servern« weiter hinten in diesem Kapitel erfahren), aber in einem großen Unternehmen mit vielen SQL Server-Instanzen werden Sie wahrscheinlich zusätzliche Ebenen der Servergruppen-Hierarchie anlegen wollen. Damit können Sie viel leichter auf Ihre SQL Server-Computer zugreifen und mit ihnen arbeiten. Sie können die folgenden Gliederungsmodelle verwenden:

- **Abteilungen oder Geschäftseinheiten** In diesem Modell spiegeln die Gruppennamen die Abteilungen oder Geschäftseinheiten wider, zu denen die SQL Server-Computer gehören. Dabei kann es z.B. die Servergruppen *Ingenieure*, *Sicherheit*, *Betrieb* und *Support* geben.
- **Geografische Verteilung** In diesem Modell spiegeln die Gruppennamen die geografische Verteilung der SQL Server-Computer wider, z.B. *Europa* und *Nordamerika*. Dabei kann es unter *Europa* wiederum zusätzliche Ebenen für *Deutschland*, *UK* und *Spanien* geben und unter *Nordamerika* Ebenen für *USA*, *Kanada* und *Mexiko*.

Abbildung 1.4 zeigt ein Beispiel. Die Untergruppen sind unter ihrer Hauptgruppe eingeordnet. Unter *Datenbankmodul* kann es z.B. die Gruppen *Ingenieure* und *Betrieb* geben und unter *Ingenieure* die Gruppen *Entwicklung*, *Test* und *Kern*.

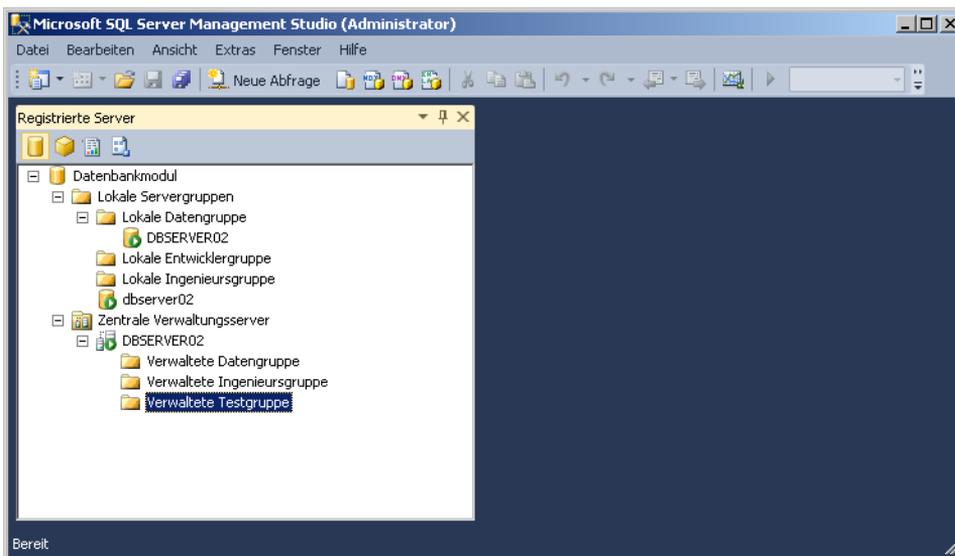


Abbildung 1.4 Servergruppen zur Gliederung von SQL Server-Bereitstellungen

Erstellen einer Servergruppe

Eine Servergruppe oder eine Untergruppe erstellen Sie wie folgt:

1. Wählen Sie in SQL Server Management Studio die Ansicht *Registrierte Server*, indem Sie **[Strg] + [Alt] + [G]** drücken. Wenn die Ansicht zuvor verborgen war, wird sie dadurch auch eingeblendet.
2. Wählen Sie über die Symbolleiste *Registrierte Server* die Gruppe der obersten Ebene aus, in der Sie die neue Gruppe anlegen möchten. Wenn Sie z.B. eine Gruppe der zweiten oder dritten Ebene unter *Datenbankmodul* anlegen möchten, wählen Sie *Datenbankmodul*.
3. Erweitern Sie nach Bedarf den Knoten für die Gruppe der obersten Ebene und den Knoten *Lokale Servergruppen*, indem Sie jeweils darauf doppelklicken. Sie sehen nun die Namen der Gruppe oberster Ebene sowie aller Servergruppen zweiter Ebene, die bereits erstellt sind. Anschließend können Sie Folgendes tun:
 - Sie können die Servergruppe einer der Gruppen erster oder zweiter Ebene hinzufügen, indem Sie auf den Gruppennamen klicken und *Neue Servergruppe* auswählen.
 - Sie können die Servergruppe einer Gruppe niedrigerer Ebene hinzufügen, indem Sie die Einträge für Servergruppen erweitern, bis die gewünschte Gruppe angezeigt wird. Rechtsklicken Sie dann auf den Gruppennamen und wählen Sie *Neue Servergruppe* aus.
4. Geben Sie im Dialogfeld *Eigenschaften für neue Servergruppe*, das Sie in Abbildung 1.5 sehen, in den vorgesehenen Feldern einen Namen und eine Beschreibung für die Gruppe ein. Klicken Sie auf *OK*.



Abbildung 1.5 Geben Sie im Dialogfeld *Eigenschaften für neue Servergruppe* einen Namen und eine Beschreibung ein

Löschen einer Servergruppe

Folgendermaßen löschen Sie eine Gruppe oder Untergruppe:

1. Wählen Sie in SQL Server Management Studio die Ansicht *Registrierte Server*, indem Sie **[Strg] + [Alt] + [G]** drücken. Wenn die Ansicht zuvor verborgen war, wird sie dadurch auch eingeblendet.
2. Wählen Sie über die Symbolleiste *Registrierte Server* die Gruppe der obersten Ebene aus, in der sich die zu löschende Gruppe befindet. Wenn Sie z.B. eine Gruppe der zweiten oder dritten Ebene unter *Datenbankmodul* entfernen möchten, wählen Sie *Datenbankmodul*.
3. Klicken Sie auf das Pluszeichen neben der Gruppe oder Untergruppe, die Sie löschen möchten. Wenn sich in der Gruppe registrierte Server befinden, verschieben Sie sie zu einer anderen Gruppe. (Die dazu erforderlichen Schritte werden weiter hinten in diesem Kapitel im Abschnitt »Verschieben eines Servers in eine andere Gruppe« erläutert.)

4. Markieren Sie den Eintrag der Gruppe oder Untergruppe.
5. Drücken Sie `[Entf]`. Wenn Sie dazu aufgefordert werden, den Vorgang zu bestätigen, klicken Sie auf *Ja*.

Bearbeiten und Verschieben von Servergruppen

Sie können verschiedene Schlüsseleigenschaften von Servergruppen bearbeiten: den Namen, die Beschreibung und den Speicherort in der Hierarchie der registrierten Server. Um den Namen oder die Beschreibung zu ändern, gehen Sie wie folgt vor:

1. Rechtsklicken Sie in der Ansicht *Registrierte Server* auf die Gruppe und wählen Sie *Eigenschaften*.
2. Geben Sie im Dialogfeld *Servergruppeneigenschaften bearbeiten* den neuen Namen und die Beschreibung ein. Klicken Sie auf *OK*.

Um eine Gruppe (und alle zugehörigen Untergruppen und Server) auf eine neue Ebene der Hierarchie zu verschieben, führen Sie folgende Schritte durch:

1. Rechtsklicken Sie in der Ansicht *Registrierte Server* auf die Gruppe, zeigen Sie auf *Tasks* und wählen Sie *Verschieben nach*.
2. Im Dialogfeld *Serverregistrierung verschieben* können Sie jetzt Folgendes tun:
 - Sie können die Gruppe in eine Gruppe der obersten Ebene verschieben, indem Sie eine Gruppe dieser Ebene auswählen. Dadurch wird Ihre Gruppe zu einer Gruppe der zweiten Ebene.
 - Sie können die Gruppe auf eine niedrigere Ebene verschieben, indem Sie eine Untergruppe auswählen, in der Sie sie platzieren möchten.
3. Klicken Sie auf *OK*.

Hinzufügen von SQL Server-Computern zu einer Gruppe

Wenn Sie einen SQL Server-Computer für die Verwendung in SQL Server Management Studio registrieren, können Sie die Gruppe für den Server auswählen. Sie können sogar eigens für diesen Server eine neue Gruppe erstellen. Im nächsten Abschnitt wird die Serverregistrierung erläutert.

Verwalten von Servern

Server und Datenbanken sind die wichtigsten Ressourcen, die Sie in SQL Server Management Studio verwalten. Wenn Sie in der Ansicht *Registrierte Server* eine Gruppe der obersten Ebene auswählen, sehen Sie die verfügbaren Servergruppen. Mit einem Doppelklick auf den Gruppennamen können Sie die Ansicht der einzelnen Gruppen erweitern und die darin befindlichen Untergruppen und Server sehen. Lokale Server werden (in den meisten Fällen) automatisch registriert. Wird ein lokaler Server nicht angezeigt, müssen Sie die lokalen Registrierungsinformationen aktualisieren. Wenn ein Remoteserver, den Sie verwalten möchten, nicht angezeigt wird, müssen Sie ihn registrieren.

Bei der Registrierung werden die aktuellen Verbindungsinformationen gespeichert und der Server zu einer Gruppe zugewiesen, sodass später über die Ansicht *Registrierte Server* leichter auf ihn zugegriffen werden kann. Nachdem Sie einen Server registriert haben, können Sie Verbindung mit ihm aufnehmen, um mit ihm zu arbeiten, und die Verbindung danach wieder trennen, indem Sie einfach auf das Serversymbol in der Ansicht *Registrierte Server* klicken. Wenn die Verbindung nicht automatisch hergestellt wird, können Sie sie

erzwingen, indem Sie mit der rechten Maustaste auf den Servereintrag klicken und dann *Neue Abfrage* auswählen (wenn Sie eine SQL-Abfrage durchführen wollen) oder *Objekt-Explorer* (wenn Sie den Server anzeigen lassen und verwalten möchten).

Sie können die Registrierung auf folgende Weisen beginnen:

- Einen Server, mit dem Sie verbunden sind, registrieren Sie im Objekt-Explorer.
- Einen neuen Server registrieren Sie in der Ansicht *Registrierte Server*.

Vorausgegangene Registrierungen können Sie auf verschiedene Weise verwalten. Es ist möglich, Folgendes zu tun:

- Sie können Registrierungsinformationen von zuvor registrierten SQL Server 2000-Servern importieren.
- Sie können Registrierungsinformationen für lokale Server aktualisieren.
- Sie können die Registrierungsinformationen durch einen Export- und einen Importvorgang von einem Computer zu einem anderen kopieren.

Registrieren eines verbundenen Servers

Jeder Server, mit dem Sie im Objekt-Explorer verbunden sind, kann einfach registriert werden. Durch die Registrierung werden die Verbindungsinformationen gespeichert, außerdem wird der Server für einen einfacheren Zugriff über die Ansicht *Registrierte Server* einer Gruppe zugewiesen. Um einen verbundenen Server zu registrieren, müssen Sie Folgendes tun:

1. Rechtsklicken Sie im Objekt-Explorer auf einen Server, mit dem Sie verbunden sind, und wählen Sie *Registrieren*, um das Dialogfeld *Neuen Server registrieren* aus Abbildung 1.6 zu öffnen.



Abbildung 1.6 Das Dialogfeld *Neuen Server registrieren*

2. Auf der Registerkarte *Allgemein* sind die aktuellen Werte für den Servernamen und die Authentifizierung bereits ausgefüllt. Die Option *Name des registrierten Servers* ist auf den Servernamen gesetzt, doch können Sie diesen Namen ändern und eine Beschreibung hinzufügen.
3. Auf der Registerkarte *Verbindungseigenschaften* können Sie die Datenbank angeben, mit der Sie Verbindung aufnehmen möchten, und Optionen für das Netzwerk und die Verbindung festlegen. Soll die Verbindung verschlüsselt werden, aktivieren Sie das Kontrollkästchen *Verbindung verschlüsseln*.
4. Um Ihre Angaben vor dem Speichern der Registrierungseinstellungen zu überprüfen, klicken Sie auf *Testen*. Falls der Test fehlschlägt, schauen Sie sich noch einmal Ihre Einstellungen an und nehmen die notwendigen Änderungen vor. Wie in Kapitel 2 gesagt, erlaubt SQL Server standardmäßig keine Remoteverbindungen. Dazu müssen Sie erst die Konfigurationseinstellungen ändern.
5. Klicken Sie auf *Speichern*, um die Serverregistrierung zu speichern.

Standardmäßig wird der Server zur Gruppe der obersten Ebene hinzugefügt. Um ihn auf eine andere Ebene der Hierarchie zu verschieben, gehen Sie wie folgt vor:

1. Rechtsklicken Sie in der Ansicht *Registrierte Server* auf den Server, zeigen Sie auf *Tasks* und wählen Sie dann *Verschieben nach*.
2. Im Dialogfeld *Serverregistrierung verschieben* können Sie den Server auf eine andere Hierarchieebene verschieben, indem Sie die Untergruppe auswählen, in der Sie ihn platzieren möchten.
3. Klicken Sie auf *OK*.

Registrieren eines neuen Servers in der Ansicht *Registrierte Server*

Um einen Server zu registrieren, müssen Sie nicht erst im Objekt-Explorer eine Verbindung mit ihm herstellen. Stattdessen können Sie ihn auch wie folgt direkt in der Ansicht *Registrierte Server* registrieren:

1. Wählen Sie in der Symbolleiste der Ansicht *Registrierte Server* den Typ des Servers aus, zu dem Sie eine Verbindung herstellen möchten, z.B. *Datenbankmodul*.
2. Erweitern Sie soweit nötig die verfügbaren Gruppen und rechtsklicken Sie dann auf die Gruppe, in der Sie den Server registrieren möchten. Wählen Sie *Neuen Server registrieren*, um das zuvor in Abbildung 1.6 gezeigte gleichnamige Dialogfeld zu öffnen.
3. Geben Sie im Feld *Servername* den vollständig qualifizierten Domänennamen oder den Hostnamen des Servers ein, auf dem SQL Server läuft, z.B. *EngDBSrv12.cpandl.com* oder *EngDBSrv12*.
4. Wählen Sie aus der Dropdownliste *Authentifizierung* den Authentifizierungstyp, entweder Windows-Authentifizierung oder SQL Server-Authentifizierung (je nachdem, welche Authentifizierungstypen Sie bei der Installation des Servers zugelassen haben). Geben Sie ggf. eine SQL Server-Anmelde-ID und ein Kennwort an.
 - **Windows-Authentifizierung** Hierbei werden Ihr Domänenkonto und Ihr Kennwort verwendet, um die Datenbankverbindung aufzubauen. Diese Art der Authentifizierung funktioniert nur, wenn die Windows-Authentifizierung aktiviert ist und Sie über die entsprechenden Rechte verfügen.
 - **SQL Server-Authentifizierung** Hierbei können Sie eine SQL Server-Anmelde-ID und ein Kennwort festlegen. Wählen Sie *Kennwort speichern*, um das Kennwort zu speichern, sodass Sie es nicht bei jedem Verbindungsaufbau neu eingeben müssen.

5. Mit den Optionen auf der Registerkarte *Verbindungseigenschaften* können Sie auch Verbindungseinstellungen festlegen. Diese Optionen erlauben Ihnen, eine Verbindung zu einer bestimmten Datenbankinstanz herzustellen und die Netzwerkkonfiguration vorzunehmen. Soll die Verbindung verschlüsselt werden, aktivieren Sie das Kontrollkästchen *Verbindung verschlüsseln*.
6. Auf der Grundlage des zuvor eingegebenen Servernamens wird der Name des registrierten Servers für Sie ausgefüllt. Ändern Sie diesen Standardnamen nur, wenn SQL Server Management Studio einen alternativen Anzeigenamen für den Server verwenden soll.
7. Klicken Sie auf *Testen*, um die Einstellungen zu überprüfen. Wenn Sie erfolgreich Verbindung mit dem Server aufnehmen können, sehen Sie eine Bestätigungsmeldung. Schlägt der Test fehl, überprüfen Sie Ihre Angaben, ändern Sie sie soweit erforderlich und testen Sie die Einstellungen erneut.
8. Klicken Sie auf *Speichern*.

Registrieren zuvor registrierter SQL Server 2000-Computer

Die Registrierungsinformationen von Servern, die schon unter SQL Server 2000 registriert wurden, können in SQL Server Management Studio importiert werden, was den Umgang mit bereits vorhandenen SQL Server 2000-Installationen vereinfacht. Wenn die SQL Server 2000-Installationen bereits auf dem Computer registriert sind, können Sie die Registrationsinformationen auf folgende Weise in eine bestimmte Servergruppe importieren:

1. Wählen Sie in der Symbolleiste der Ansicht *Registrierte Server* den Typ des Servers aus, den Sie registrieren möchten, z.B. *Datenbankmodul*.
2. Rechtsklicken Sie auf den Eintrag *Lokale Servergruppen*, zeigen Sie auf *Tasks* und wählen Sie *Zuvor registrierte Server*.
3. Die verfügbaren Registrierungsinformationen für SQL Server 2000-Computer werden importiert. Wenn dabei eine Fehlermeldung erscheint, kann es sein, dass Sie nicht lokal an dem Computer angemeldet sind, auf dem die Server zuvor registriert wurden.

Aktualisieren der Registrierungsinformationen für lokale Server

Lokale Server werden (in den meisten Fällen) automatisch registriert. Wenn Sie auf dem lokalen Computer SQL Server-Instanzen hinzugefügt oder entfernt haben und diese Änderungen nicht angezeigt werden, müssen Sie die lokale Serverregistrierung aktualisieren, damit alle zurzeit eingerichteten lokalen Serverinstanzen in SQL Server Management Studio angezeigt werden.

Um die Registrierungsinformationen für lokale Server zu aktualisieren, gehen Sie wie folgt vor:

1. Wählen Sie in der Symbolleiste der Ansicht *Registrierte Server* den Typ des Servers aus, den Sie registrieren möchten, z.B. *Datenbankmodul*.
2. Rechtsklicken Sie auf den Eintrag *Lokale Servergruppen*, zeigen Sie auf *Tasks* und wählen Sie *Lokale Server registrieren*.

Kopieren von Servergruppen und lokalen Registrierungsinformationen

Nachdem Sie Server in SQL Server Management Studio registriert und in einer Gruppenhierarchie platziert haben, kann es vorkommen, dass Sie die Registrierungsinformationen und die Servergruppenstruktur auch auf einem anderen Computer verwenden möchten. Mit SQL Server Management Studio können Sie die Registrierungsinformationen mit einem Import/Export-Vorgang von einem Rechner auf den anderen kopieren. Dabei können Sie entscheiden, ob Sie die Benutzernamen und Kennwörter mitkopieren oder nicht.

Um Registrierungs- und Gruppeninformationen in eine Datei auf einem Computer zu exportieren und sie dann auf einem anderen Rechner zu importieren, gehen Sie wie folgt vor:

1. Starten Sie SQL Server Management Studio auf dem Computer mit den Informationen über die Registrierung und die Gruppenhierarchie, die Sie kopieren möchten.
2. Rufen Sie die Ansicht *Registrierte Server* auf, indem Sie **[Strg] + [Alt] + [G]** drücken.
3. Wählen Sie in der Symbolleiste der Ansicht *Registrierte Server* den Typ des Servers aus, den Sie registrieren möchten, z.B. *Datenbankmodul*.
4. Rechtsklicken Sie auf den Eintrag *Lokale Servergruppen* und wählen Sie *Exportieren*, um das Dialogfeld *Registrierte Server exportieren* aus Abbildung 1.7 zu öffnen.

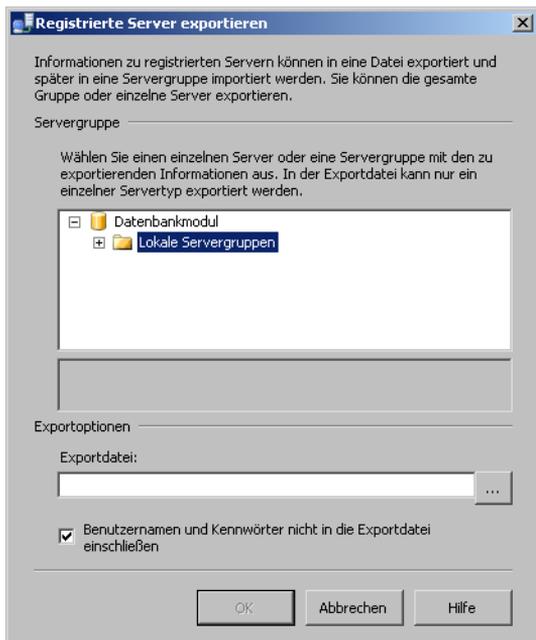


Abbildung 1.7 Das Dialogfeld *Registrierte Server exportieren*

5. Wählen Sie unter *Servergruppe* den Punkt aus, an dem der Exportvorgang beginnen soll. Sie können auf jeder Ebene der Gruppenhierarchie mit dem Kopieren von Registrierungsinformationen beginnen:
 - Um die Struktur einer Gruppe der obersten Ebene, all ihre Untergruppen und alle Registrierungsinformationen der damit verbundenen Server zu kopieren, wählen Sie die Gruppe der obersten Ebene aus.

- Um die Struktur einer Untergruppe, all ihre Untergruppen und alle Registrierungsinformationen der damit verbundenen Server zu kopieren, wählen Sie die Untergruppe aus.
 - Um die Registrierungsinformationen eines einzelnen Servers zu kopieren, wählen Sie den Server aus.
6. Die Struktur der Servergruppe und die Registrierungsinformationen werden in eine Datei mit der Endung REGSRVR exportiert. Standardmäßig wird diese Datei im Ordner *Dokumente* des zurzeit angemeldeten Benutzers angelegt. Unter *Exportoptionen* können Sie einen Namen für die Datei anlegen, z.B. *CurrentDBConfig*.

TIPP

Wenn Sie die Exportdatei in einer sicheren Netzwerkfreigabe platzieren, können Sie von dem Computer aus darauf zugreifen, auf den Sie die Registrierungsinformationen kopieren möchten. Anderenfalls müssen Sie die Datei auf diesen Rechner kopieren.

7. Standardmäßig werden die aktuellen Authentifizierungsinformationen für Serververbindungen nicht exportiert. Wenn Sie die Benutzernamen und Kennwörter exportieren möchten, müssen Sie das Kontrollkästchen *Benutzernamen und Kennwörter nicht in die Exportdatei einschließen* deaktivieren.
8. Klicken Sie auf *OK*. Wenn der Export erfolgreich war, wird Ihnen das in einer Meldung bestätigt. Klicken Sie in dem Meldungsfeld auf *OK*. Wenn ein Fehler auftrat, notieren Sie sich die Fehlermeldung und beheben Sie das Problem.
9. Starten Sie SQL Server Management Studio auf dem Computer, auf den Sie die Servergruppen- und Registrierungsinformationen kopieren möchten. Wenn Sie die Exportdatei nicht auf einer sicheren Netzwerkfreigabe gespeichert haben, müssen Sie sie jetzt auf diesen Rechner kopieren.
10. Rufen Sie die Ansicht *Registrierte Server* auf, indem Sie **[Strg] + [Alt] + [G]** drücken.
11. Wählen Sie in der Symbolleiste der Ansicht *Registrierte Server* den Typ des Servers aus, den Sie registrieren möchten, z.B. *Datenbankmodul*.
12. Rechtsklicken Sie auf den Eintrag *Lokale Servergruppen* und wählen Sie *Importieren*, um das Dialogfeld *Registrierte Server importieren* aus Abbildung 1.8 zu öffnen.

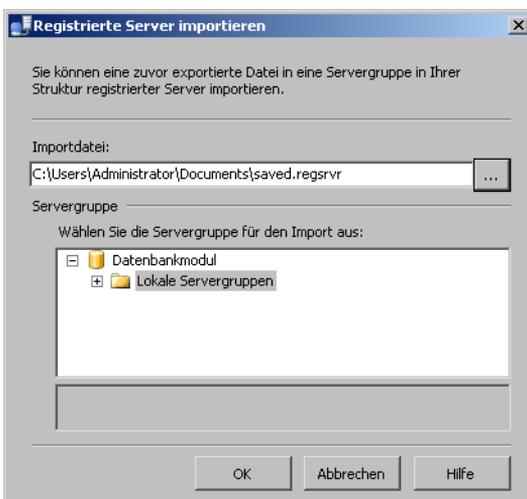


Abbildung 1.8 Das Dialogfeld *Registrierte Server importieren*

13. Klicken Sie im Dialogfeld auf die Schaltfläche rechts neben dem Textfeld *Importdatei* und wählen Sie in dem daraufhin angezeigten Dialogfeld *Öffnen* die Datei aus, die Sie importieren möchten.
14. Wählen Sie unter *Servergruppe* die Servergruppe aus, unter der die importierten Gruppen und Server erstellt werden sollen.
15. Klicken Sie auf *OK*. Wenn der Import erfolgreich war, wird Ihnen das in einer Meldung bestätigt. Klicken Sie in dem Meldungsfeld auf *OK*. Wenn ein Fehler auftrat, notieren Sie sich die Fehlermeldung und beheben Sie das Problem.

Bearbeiten der Registrierungseigenschaften

Sie können die Registrierungseigenschaften eines Servers jederzeit ändern, indem Sie in der Ansicht *Registrierte Server* von SQL Server Management Studio mit der rechten Maustaste auf den Eintrag für den Server klicken und *Eigenschaften* wählen. Nehmen Sie Änderungen dann im Dialogfeld *Serverregistrierungseigenschaften bearbeiten* vor. Die einzige Eigenschaft, die Sie nicht ändern können, ist der Servertyp. Testen Sie die Einstellungen, bevor Sie sie speichern.

Verbinden mit einem Server

Nachdem Sie einen Server registriert haben, können Sie auf sehr einfache Weise die Verbindung zu ihm herstellen. Rechtsklicken Sie auf den Servereintrag in der Ansicht *Registrierte Server* von SQL Server Management Studio und wählen Sie *Neue Abfrage* (wenn Sie eine SQL-Abfrage durchführen wollen) oder *Objekt-Explorer* (wenn Sie den Server anzeigen lassen und verwalten möchten). Sie können auch auf den Servereintrag doppelklicken, um eine Verbindung herzustellen und im Objekt-Explorer mit dem Server zu arbeiten.

HINWEIS

SQL Server Management Studio stellt die Verbindungen zu anderen SQL Server-Computern über das Netzwerkprotokoll her, das in den Registrierungseigenschaften festgelegt ist. Wenn Sie das Netzwerkprotokoll oder den gesamten Remotezugriff deaktiviert haben, können Sie in SQL Server Management Studio nicht auf diesen Server zugreifen. In diesem Fall müssen Sie die erforderlichen Änderungen an den Registrierungseigenschaften oder in der Oberflächenkonfiguration vornehmen. In Kapitel 2 wird die Oberflächenkonfiguration erklärt.

Trennen der Verbindung zu einem Server

Wenn Sie mit der Arbeit an einem Server fertig sind, können Sie die Verbindung zu ihm trennen. Dadurch wird die Kommunikation mit dem Server eingestellt. Klicken Sie hierzu im Objekt-Explorer von SQL Server Management Studio mit der rechten Maustaste auf den Eintrag für den Server und wählen Sie im Kontextmenü *Trennen*.

Verschieben eines Servers in eine andere Gruppe

Um den Server in eine andere Gruppe zu verschieben, gehen Sie wie folgt vor:

1. Rechtsklicken Sie in der Ansicht *Registrierte Server* auf den Server und wählen Sie *Verschieben* aus dem Kontextmenü, um das Dialogfeld *Serverregistrierung verschieben* zu öffnen.

2. Erweitern Sie im Dialogfeld *Serverregistrierung verschieben* den Eintrag *Lokale Servergruppen*, um die Liste der Untergruppen einzusehen. Soweit nötig, erweitern Sie auch die Untergruppen. Sie können jetzt Folgendes tun:
 - Sie können den Server in die Gruppe der obersten Ebene verschieben, indem Sie sie auswählen. Dadurch wird der Server ein Mitglied dieser Gruppe.
 - Sie können den Server auf eine andere Ebene verschieben, indem Sie die Untergruppe auswählen, in der Sie ihn platzieren möchten.
3. Klicken Sie auf *OK*.

Löschen einer Serverregistrierung

Wenn Sie einen Servernamen ändern oder einen Server entfernen, können Sie die Serverregistrierung in SQL Server Management Studio löschen, damit dieses Programm nicht mehr versucht, eine Verbindung zu einem Server herzustellen, auf den nicht mehr zugegriffen werden kann. Klicken Sie in der Ansicht *Registrierte Server* mit der rechten Maustaste auf den Eintrag für den Server und wählen Sie *Löschen*. Wenn Sie aufgefordert werden, diesen Schritt zu bestätigen, klicken Sie auf *Ja*.

Verwenden von Windows PowerShell zur Verwaltung von SQL Server

Die grafischen Verwaltungswerkzeuge bieten Ihnen jegliche Möglichkeiten, die Sie zur Arbeit mit SQL Server benötigen. Es gibt jedoch Gelegenheiten, bei denen Sie lieber an der Befehlszeile arbeiten möchten, z.B. bei einer Windows Server 2008 R2 Core-Installation. Um auch diese Bedürfnisse zu erfüllen, umfasst SQL Server 2012 den SQL Server-Anbieter für Windows PowerShell (kurz »SQL Server PowerShell« genannt). Damit Sie über Windows PowerShell mit SQL Server arbeiten können, müssen Sie als Erstes ein Befehlszeilenfenster oder eine Windows PowerShell-Eingabeaufforderung öffnen und dann SQL Server PowerShell starten, indem Sie dort *sqlps* eingeben.

Windows PowerShell führt das Prinzip so genannter *Cmdlets* (ausgesprochen Commandlet) ein – kleinstmöglicher Funktionseinheiten innerhalb von PowerShell. Bei Cmdlets wird nicht zwischen Groß- und Kleinschreibung unterschieden. Unter anderem enthält SQL Server PowerShell die folgenden Cmdlets:

- **Backup-SQLDatabase** Führt Sicherungsvorgänge für SQL Server-Datenbanken durch.
- **Convert-UrnToPath** Wandelt den einheitlichen Ressourcennamen (Uniform Resource Name, URN) eines SQL Server Management-Objekts in einen SQL Server-Anbieterpfad um. Der URN gibt den Speicherort eines Verwaltungsobjekts innerhalb der SQL Server-Objekthierarchie an. Enthält der URN-Pfad Zeichen, die nicht von Windows PowerShell unterstützt werden, werden diese automatisch kodiert.
- **Decode-SQLName** Gibt bei Übergabe eines kodierten Bezeichners einen unkodierten SQL Server-Bezeichner zurück.
- **Encode-SQLName** Kodiert Sonderzeichen in SQL Server-Bezeichnern und -Namenspfaden in ein für PowerShell-Pfade verwertbares Format. Folgende Sonderzeichen werden durch dieses Cmdlet kodiert: `\:/%<>?*?[]|`. Wenn Sie diese Zeichen nicht verschlüsseln, müssen Sie sie in einfache Anführungszeichen (') setzen.

- **Invoke-PolicyEvaluation** Überprüft die aktuellen Verwaltungsrichtlinien der SQL Server-Instanz. Dieses Cmdlet gibt standardmäßig den Status der Kompatibilität zurück, erzwingt sie jedoch nicht. Um die Kompatibilität durchzusetzen, setzen Sie *-AdHocPolicyEvaluationMode* auf *Configure*.
- **Invoke-SQLCmd** Startet ein Transact-SQL- (TSQL) oder XQuery-Skript mit Befehlen, die durch das Dienstprogramm SQLCMD unterstützt werden. Dieses Cmdlet legt standardmäßig weder SQLCMD-Variablen fest noch gibt es Meldungen zurück. Nur ein Teil der SQLCMD-Befehle kann genutzt werden.
- **Restore-SQLDatabase** Führt Wiederherstellungsvorgänge für SQL Server-Datenbanken durch.

Geben Sie *get-help Cmdletname-detailed* ein, um ausführliche Informationen über ein bestimmtes Cmdlet zu erhalten, und *get-help sqlpserver | more*, um ausführliche Informationen über den Anbieter der SQL Server-Funktionen für Windows PowerShell zu erhalten.

Praxistipp

Das Dienstprogramm *sqlps* können Sie auf jedem Computer einsetzen, auf dem SQL Server oder die Befehlszeilenwerkzeuge installiert sind. Das Programm startet eine Windows PowerShell-Sitzung mit dem importierten SQL Server-Anbieter, sodass Sie dessen Cmdlets verwenden und mit SQL Server-Instanzen arbeiten können. Bei der Verwendung von Windows PowerShell oder Skripts können Sie das SQLPS-Modul importieren, um den SQL Server-Anbieter zu laden, wodurch automatisch die erforderlichen Assemblies geladen werden und die Umgebung initialisiert wird. Früher war es notwendig, ein Initialisierungsskript einzusetzen, doch inzwischen ist das nicht mehr erforderlich, sofern Sie vor dem Zugriff auf die SQL Server-Instanz das SQLPS-Modul importieren. Um die besten Ergebnisse zu erzielen, sollten Sie das Modul mit dem folgenden Befehl importieren:

```
Import-Module "sqlps" -DisableNameChecking
```

Sie können mit Cmdlets arbeiten, indem Sie Befehle entweder direkt an der Befehlszeile oder aus Skripts heraus ausführen. Jegliche Befehle und Cmdlets, die sich an der Befehlszeile von Windows PowerShell ausführen lassen, können Sie in ein Skript einfügen, indem Sie den entsprechenden Befehltext in eine Datei kopieren und diese mit der Dateierweiterung *PS1* speichern. Anschließend können Sie das Skript auf dieselbe Weise ausführen wie einen Befehl oder ein Cmdlet. Wenn Sie mit Windows PowerShell arbeiten, kann es allerdings sein, dass das aktuelle Verzeichnis nicht Teil des Umgebungspfades ist, sodass Sie – wenn Sie ein Skript im aktuellen Verzeichnis ausführen – vor dem Skriptnamen die Zeichenfolge *./* eingeben müssen:

```
./runtasks
```

Die aktuelle Ausführungsrichtlinie für SQL Server PowerShell steuert, ob und wie Skripts ausgeführt werden können. Zwar hängt die Standardkonfiguration von der jeweiligen Edition sowie vom verwendeten Betriebssystem ab, jedoch können Sie die Ausführungsrichtlinie schnell ermitteln, indem Sie den Befehl *get-executionpolicy* an der Befehlszeile von PowerShell eingeben.

Um die Ausführungsrichtlinie so einzurichten, dass alle Skripts eine vertrauenswürdige Signatur aufweisen müssen, um ausgeführt zu werden, geben Sie den folgenden Befehl ein:

```
set-executionpolicy allsigned
```

Um zu erreichen, dass aus dem Web heruntergeladene Skripts nur dann ausgeführt werden, wenn sie die Signatur einer vertrauenswürdigen Quelle tragen, geben Sie Folgendes ein:

```
set-executionpolicy remotesigned
```

Der folgende Befehl bewirkt, dass alle Skripts unabhängig davon ausgeführt werden, ob sie eine digitale Signatur haben, sodass Sie in einer unbeschränkten Umgebung arbeiten können:

```
set-executionpolicy unrestricted
```

Für Verwaltungsaufgaben an der Eingabeaufforderung von Windows PowerShell können Sie T-SQL- oder XQuery-Skripts ausführen, die von SQLCMD unterstützte Befehle enthalten. Verwenden Sie dazu *Invoke-Sqlcmd*. Dabei ist die gesamte T-SQL- und XQuery-Syntax zulässig, die auch im Datenbankmodul verwendet werden kann, allerdings sind standardmäßig keine Skriptvariablen festgelegt. *Invoke-Sqlcmd* akzeptiert auch die weiter hinten in diesem Kapitel in Tabelle 1.3 aufgeführten SQLCMD-Befehle. Standardmäßig werden die Ergebnisse in Tabellenform ausgegeben, wobei das erste Resultset automatisch angezeigt wird und die folgenden nur dann, wenn Sie über dieselben Spalten verfügen.

Die grundlegende Syntax, die Sie für *Invoke-Sqlcmd* in den meisten Fällen einsetzen werden, lautet:

```
Invoke-Sqlcmd [-ServerInstance Serverstring_oder_Objekt]  
[-Database Datenbankname] [-EncryptConnection ]  
[-Username Benutzername] [-Password Kennwort] [[-Query] Abfragestring]  
[-DedicatedAdministratorConnection]  
[-InputFile Dateipfad] [ | Out-File -filepath Dateipfad]
```

Dieser Befehl hat folgende Parameter:

- **-Database** Nennt den Namen der Datenbank, mit der Sie arbeiten möchten. Welche Datenbank verwendet wird, wenn Sie diesen Parameter nicht angeben, hängt davon ab, ob der aktuelle Pfad sowohl den Ordner SQLSERVER:\SQL als aus einen Datenbanknamen enthält. Sind beide angegeben, nimmt *Invoke-Sqlcmd* Verbindung mit der im Pfad genannten Datenbank auf, anderenfalls mit der Standarddatenbank für die aktuelle Anmelde-ID.

HINWEIS

Mit *-IgnoreProviderContext* erzwingen Sie eine Verbindung zu der festgelegten Standarddatenbank für die aktuelle Anmelde-ID.

- **-DedicatedAdministratorConnection** Sorgt dafür, dass eine dedizierte Administratorverbindung (Dedicated Administrator Connection, DAC) verwendet wird, um eine Verbindung zu erzwingen, wenn ansonsten keine möglich wäre.
- **-EncryptConnection** Aktiviert die SSL-Verschlüsselung (Secure Sockets Layer) für die Verbindung.
- **-InputFile** Gibt den vollständigen Pfad zu einer Datei an, die als Eingabe der Abfrage verwendet werden soll. Diese Datei kann T-SQL-Anweisungen, XQuery-Anweisungen, SQLCMD-Befehle und Skriptvariablen enthalten. Im Dateipfad und -namen sind keine Leerzeichen erlaubt.

- **-Password** Legt das Kennwort für die mit *-Username* angegebene Anmelde-ID der SQL Server-Authentifizierung fest.
- **-Query** Definiert eine oder mehrere Abfragen, die ausgeführt werden sollen. Dabei kann es sich um T-SQL-Abfragen, XQuery-Anweisungen und SQLCMD-Befehle handeln. Mehrere Abfragen werden durch Semikolons getrennt.

TIPP

Den Befehl SQLCMD GO müssen Sie nicht verwenden. Maskieren Sie alle doppelten Anführungszeichen im String und verwenden Sie für Bezeichner eckige Klammern wie in *[EmpTable]* anstelle von Anführungszeichen wie in »*Emp-Table*«. Um sicherzustellen, dass die Ausgabe zurückgegeben wird, fügen Sie den Parameter *-Verbose* hinzu. Dieser Parameter ist in allen Cmdlets verfügbar.

- **-ServerInstance** Gibt den Namen der Datenbankmodul-Instanz an, mit der Sie arbeiten möchten. Für Standardinstanzen müssen Sie lediglich den Computernamen nennen, also z.B. *DbServer23*. Bei benannten Instanzen verwenden Sie das Format *Computername\Instanzname*, also etwa *DbServer23\EmployeeDb*.
- **-Username** Legt die Anmelde-ID für die SQL Server-Authentifizierung an einer Instanz des Datenbankmoduls fest. Für diese Anmelde-ID müssen Sie auch das Kennwort angeben.

HINWEIS

Standardmäßig versucht der Befehl *Invoke-Sqlcmd* eine Verbindung mit Windows-Authentifizierung einzugehen, indem er das Windows-Konto verwendet, unter dem die Windows PowerShell-Sitzung ausgeführt wird. Um stattdessen eine Verbindung mit SQL Server-Authentifizierung zu nutzen, geben Sie den Benutzernamen und das Kennwort für die gewünschte SQL Server-Anmelde-ID an.

Mit diesen Kenntnissen können Sie die folgenden T-SQL-Anweisungen:

```
USE OrderSystem;
GO
SELECT * FROM Inventory.Product
ORDER BY Name ASC
GO
```

durch den folgenden Windows PowerShell-Befehl ersetzen:

```
Invoke-Sqlcmd -Query "SELECT * FROM Inventory.Product; ORDER BY Name ASC"
-ServerInstance "DbServer23\OrderSystem"
```

Es ist auch möglich, die Befehle aus einem Skript zu lesen, wie Beispiel 1.1 zeigt:

Inhalt des Skripts SqlCmd.sql

```
SELECT * FROM Inventory.Product
ORDER BY Name ASC
```

Befehl zum Ausführen des Skripts

```
Invoke-Sqlcmd -InputFile "C:\Scripts\SqlCmd.sql"
```

Beispiel 1.1 SQL-Befehlsskript

Die Tatsache, dass die SQL Server-Unterstützung über einen Anbieter erfolgt, ist bei der Arbeit mit Windows PowerShell von großer Wichtigkeit. Die von Anbietern bereitgestellten Daten erscheinen als Laufwerk, das Sie durchsuchen können. Eine Möglichkeit dazu besteht darin, den Speicherort relativ zum Anbieterlaufwerk *SqlServer*: festzulegen. Die Spitze der bereitgestellten Hierarchie wird durch den Ordner *SQL* dargestellt. Es folgen ein Ordner mit dem Computernamen und einer mit dem Instanznamen. Dadurch können Sie durch folgende Eingabe zum obersten Ordner der Standardinstanz wechseln:

```
Set-Location SQLSERVER:\SQL\DbServer23\Default
```

Anschließend können Sie die verfügbaren Datenbankstrukturen bestimmen, indem Sie *Get-ChildItem* (oder einen der Aliase dieses Befehls wie *ls* oder *dir*) eingeben. Um zu Anmeldungen, Triggern, Endpunkten, Datenbanken und anderen Strukturen zu gelangen, setzen Sie den Speicherort auf den Namen des zugehörigen Ordners. Beispielsweise können Sie *Set-Location Databases* verwenden und dann *Get-ChildItem* eingeben, um eine Liste der verfügbaren Datenbanken in der ausgewählten Instanz abzurufen. Wenn Sie den vollständigen Pfad kennen, können Sie auch direkt auf den gewünschten Ort zugreifen, wie das folgende Beispiel zeigt:

```
Set-Location SQLSERVER:\SQL\DbServer23\Default\Databases\OrderSystem
```

Hiermit wechseln Sie zu der Struktur für die Datenbank *OrderSystem* auf der Standardinstanz von *DbServer23*. Wollen Sie herausfinden, welche Tabellen in dieser Datenbank verfügbar sind, geben Sie Folgendes ein:

```
Get-ChildItem Tables
```

Auch dieser Befehl ist möglich:

```
Set-Location TablesGet-ChildItem
```

Um SQL Server 2012 von einem Computer aus zu verwalten, auf dem SQL Server nicht ausgeführt ist, müssen Sie die Verwaltungstools installieren. Wählen Sie im SQL Server-Installationscenter *Installation* und klicken Sie dann auf *Neuinstallation oder Hinzufügen von Funktionen zu einer vorhandenen Installation*. Folgen Sie den Anweisungen des Assistenten. Wählen Sie auf der Seite *Featureauswahl* die Option *Verwaltungstools – Einfach*, um Management Studio, SQLCMD und den SQL Server-Anbieter für Windows PowerShell zu installieren.

Zur Remoteverwaltung über Windows PowerShell müssen sowohl Windows Remote Management (WinRM) als auch Windows PowerShell installiert und mit dem Assistenten zum Hinzufügen von Funktionen verfügbar gemacht sein. Außerdem müssen Sie auf dem Verwaltungscomputer und dem SQL Server-Computer Remotebefehle aktivieren.

Mit den folgenden Schritten prüfen Sie die Verfügbarkeit von WinRM und richten Windows PowerShell für die Remoteverwendung ein:

1. Klicken Sie auf *Start*, *Alle Programme*, *Zubehör* und *Windows PowerShell*. Starten Sie dann Windows PowerShell als Administrator, indem Sie mit der rechten Maustaste auf die zugehörige Verknüpfung klicken und *Als Administrator ausführen* auswählen.

2. Der WinRM-Dienst ist standardmäßig für den manuellen Start eingerichtet. Sie müssen den Starttyp in *Automatisch* ändern und den Dienst auf jedem Computer starten, mit dem Sie arbeiten möchten. Ob der WinRM-Dienst läuft, können Sie an der PowerShell-Eingabeaufforderung mit dem folgenden Befehl überprüfen:

```
get-service winrm
```

Wie das folgende Beispiel zeigt, sollte die Eigenschaft *Status* in der Ausgabe den Wert *Running* aufweisen:

Status	Name	DisplayName
Running	WinRM	Windows Remote Management

Wenn der Dienst beendet ist, geben Sie den folgenden Befehl ein, um ihn zu starten und so einzurichten, dass er in Zukunft automatisch gestartet wird:

```
set-service -name winrm -startuptype automatic -status running
```

3. Um Windows PowerShell für die Remoteverwaltung einzurichten, geben Sie folgenden Befehl ein:

```
Enable-PSRemoting -force
```

Die Remoteverwaltung können Sie nur dann aktivieren, wenn der Computer mit einer Domäne oder einem privaten Netzwerk verbunden ist. Sind die Computer an ein öffentliches Netzwerk angeschlossen, müssen Sie diese Anbindung trennen und eine Verbindung zu einer Domäne oder einem privaten Netzwerk herstellen, bevor Sie diesen Schritt durchführen können. Wenn mindestens eine der Computerverbindungen von einem öffentlichen Typ ist, in Wirklichkeit aber eine Verbindung zu einer Domäne oder einem privaten Netzwerk besteht, müssen Sie zunächst den Verbindungstyp im Netzwerk- und Freigabe-center ändern.

In vielen Fällen ist es möglich, mit Remotecomputern in anderen Domänen zu arbeiten. Ist diese Domäne jedoch nicht vertrauenswürdig, kann der Remotecomputer Ihre Anmeldeinformationen eventuell nicht authentifizieren. Um die Authentifizierung zu ermöglichen, müssen Sie den Remotecomputer in WinRM zur Liste der vertrauenswürdigen Hosts für den lokalen Computer hinzufügen. Geben Sie dazu Folgendes ein:

```
winrm s winrm/config/client '@{TrustedHosts="RemoteComputer"}'
```

wobei Sie *RemoteComputer* durch den tatsächlichen Namen des betreffenden Rechners ersetzen:

```
winrm s winrm/config/client '@{TrustedHosts="DbServer23"}'
```

Bei der Arbeit mit Computern in Arbeits- und Heimnetzgruppen müssen Sie HTTPS als Transportprotokoll verwenden oder den Remotecomputer zur Konfigurationseinstellung *TrustedHosts* hinzufügen. Wenn zu einem Remotehost keine Verbindung möglich ist, können Sie überprüfen, ob der Dienst dort vorhanden ist und Anforderungen entgegennimmt, indem Sie auf dem Remotehost den folgenden Befehl eingeben:

```
winrm quickconfig
```

Dieser Befehl analysiert und konfiguriert den WinRM-Dienst. Wenn dieser Dienst korrekt eingerichtet ist, erhalten Sie folgende Ausgabe:

```
WinRM ist bereits zum Empfangen von Anforderungen auf diesem Computer konfiguriert.  
WinRM ist bereits für die Remoteverwaltung auf diesem Computer eingerichtet.
```

Ist WinRM dagegen nicht richtig eingerichtet, werden Fehlermeldungen angezeigt. Sie müssen dann verschiedene Anzeigen bestätigen, um die Remoteverwaltung automatisch einzurichten. Wenn dieser Vorgang abgeschlossen ist, sollte WinRM korrekt konfiguriert sein. Denken Sie daran, dass Sie die Remoteverwaltung sowohl auf dem Datenbankserver als auch auf dem Verwaltungscomputer einrichten müssen.

Starten, Beenden und Konfigurieren des SQL Server-Agents

Der SQL Server-Agent wird als Dienst ausgeführt und zum Planen von Aufträgen, Warnungen und anderen automatisierten Aufgaben verwendet. Nachdem Sie automatisierte Aufgaben geplant haben, sollten Sie dafür sorgen, dass der SQL Server-Agent automatisch beim Systemstart ausgeführt wird. Dadurch wird sichergestellt, dass die Aufgaben wie erwartet ausgeführt werden. Mit dem SQL Server-Konfigurations-Manager können Sie den zugehörigen Dienst (*Instanzname*) wie den SQL Server-Dienst steuern. Einzelheiten hierzu finden Sie im Abschnitt »Konfigurieren der SQL Server-Dienste« von Kapitel 2.

Um den SQL Server-Agent zu konfigurieren, verwenden Sie SQL Server Management Studio. Die Agent-Konfiguration wird in Kapitel 10, »Automatisierung und Wartung von SQL Server 2012«, detailliert erläutert. Gehen Sie hierzu grundsätzlich folgendermaßen vor:

1. Verbinden Sie sich mit dem Datenbankmodul auf dem Server, den Sie konfigurieren möchten. Sie können dies in der Ansicht *Registrierte Server* erledigen, indem Sie auf den Servereintrag doppelklicken, oder im Objekt-Explorer. Klicken Sie dort auf *Verbinden* und wählen Sie *Datenbankmodul*, um das Dialogfeld *Verbindung mit Server herstellen* zu öffnen.
2. Rechtsklicken Sie auf den Knoten *SQL Server-Agent* und wählen Sie *Eigenschaften* aus dem Kontextmenü. Sie können den SQL Server-Agent jetzt konfigurieren. Sollte dieser Dienst nicht ausgeführt werden, müssen Sie ihn starten, um seine Eigenschaften zu bearbeiten.
3. Über das Kontextmenü des SQL Server-Agents können Sie auch den SQL Server-Agent-Dienst verwalten. Wählen Sie je nach Aufgabenstellung *Starten*, *Beenden* oder *Neu starten*.

Starten, Beenden und Konfigurieren von Microsoft Distributed Transaction Coordinator

Microsoft Distributed Transaction Coordinator (MSDTC) ist ein Transaktions-Manager, über den Clientanwendungen mit mehreren Datenquellen in einer Transaktion arbeiten können.

Wenn eine verteilte Transaktion mehrere Server umfasst, koordinieren die Server die Verwaltung der Transaktion mit MSDTC. Schließt eine verteilte Transaktion mehrere Datenbanken auf einem Server ein, verwaltet SQL Server die Transaktion intern.

SQL Server-Anwendungen können MSDTC direkt aufrufen, um eine explizite verteilte Transaktion zu starten. Verteilte Transaktionen können durch folgende Methoden auch implizit gestartet werden:

- Aufrufen von gespeicherten Prozeduren auf Remoteservern unter SQL Server
- Aktualisieren von Daten in mehreren OLE DB-Datenquellen
- Eintragen von Remoteservern in eine Transaktion

Wenn Sie mit Transaktionen in einem der zuvor aufgelisteten Szenarien arbeiten, sollten Sie MSDTC auf dem Server ausführen und beim Serverstart automatisch aufrufen. MSDTC wird wie SQL Server als Dienst ausgeführt. Im Gegensatz zum SQL Server-Dienst kann nur eine Instanz des MSDTC-Dienstes auf einem Computer ausgeführt werden. Die Anzahl der verfügbaren Datenbankserver-Instanzen ist dabei nicht von Bedeutung. Dies bedeutet, dass alle Instanzen von SQL Server auf einem Computer den gleichen Transaktionskoordinator verwenden.

Sie können den aktuellen Status von MSDTC in SQL Server Management Studio ablesen, indem Sie eine Verbindung mit dem Datenbankmodul des Servers herstellen. Erweitern Sie im Objekt-Explorer den Server und den Knoten *Verwaltung*. Wenn der Dienst ausgeführt wird, sehen Sie einen grünen Kreis mit einem nach rechts weisenden Dreieck darin (ähnlich einer Wiedergabe-Taste). Ist der Dienst beendet, wird ein roter Kreis mit einem Viereck darin angezeigt (ähnlich einer Stopp-Taste). Den MSDTC-Dienst können Sie über die Computerverwaltung steuern. Gehen Sie dazu wie folgt vor:

1. Klicken Sie auf die *Start*-Schaltfläche, zeigen Sie auf *Alle Programme* und *Verwaltung* und wählen Sie *Computerverwaltung*.
2. Standardmäßig werden Sie mit dem lokalen Computer verbunden. Um Verbindung mit einem Remotecomputer aufzunehmen, rechtsklicken Sie auf den Knoten *Computerverwaltung* und wählen *Verbindung mit anderem Computer herstellen*. Wählen Sie im Dialogfeld *Computer auswählen* die Option *Anderer Computer* und geben Sie den Namen des Rechners ein. Dies kann in Form des Hostnamens geschehen, also z.B. als *EngDBSvr12*, oder als vollständig qualifizierter Domänenname wie *EngDBSvr12.cpandl.com*.
3. Erweitern Sie *Dienste und Anwendungen* und wählen Sie *Dienste*. Rechtsklicken Sie auf *Distributed Transaction Coordinator* und wählen Sie *Eigenschaften*. Jetzt können Sie MSDTC verwalten.

Einrichten des Startverhaltens von SQL Server

Das SQL Server-Datenbankmodul weist zwei Betriebsmodi auf. Es kann als Dienst und als Befehlszeilenanwendung (*SQLServr.exe*) ausgeführt werden. Letztere verwenden Sie, wenn Sie die Ursache von Fehlern suchen oder Konfigurationseinstellungen im Einzelbenutzermodus ändern.

Einrichten oder Verhindern des automatischen Starts

In Kapitel 2 erfahren Sie, wie Sie den SQL Server-Konfigurations-Manager verwenden können, um den SQL Server-Dienst (MSSQLSERVER), verwandte Dienste für andere Instanzen des Datenbankmoduls und andere Dienste von SQL Server zu verwalten. Jeder dieser Dienste lässt sich für den automatischen Start einrichten, Sie können einen automatischen Start aber auch verhindern. Dazu gehen Sie wie folgt vor:

1. Starten Sie den SQL Server-Konfigurations-Manager mit einer der folgenden Methoden:
 - Melden Sie sich lokal oder über das Netzwerk am Datenbankserver an und starten Sie den SQL Server-Konfigurations-Manager, indem Sie auf die *Start*-Schaltfläche klicken, auf *Alle Programme*, *Microsoft SQL Server 2012* und *Konfigurationstools* zeigen und schließlich *SQL Server-Konfigurations-Manager* auswählen.
 - Öffnen Sie in SQL Server Management Studio die Ansicht *Registrierte Server*, indem Sie **[Strg] + [Alt] + [G]** drücken. Wählen Sie über die Symbolleiste *Registrierte Server* die Gruppe der obersten Ebene aus und erweitern Sie die Gruppenknoten, indem Sie darauf doppelklicken. Rechtsklicken Sie auf den Servereintrag und wählen Sie *SQL Server-Konfigurations-Manager*.
2. Markieren Sie den Knoten *SQL Server-Dienste*. Rechtsklicken Sie auf den SQL Server-Dienst, den Sie automatisch starten lassen möchten, und wählen Sie *Eigenschaften*. Sie haben jetzt folgende Möglichkeiten:
 - **Den automatischen Start aktivieren** Setzen Sie auf der Registerkarte *Dienste* den Startmodus auf *Automatisch*. Wenn als Status *Beendet* erscheint, klicken Sie auf der Registerkarte *Anmelden* auf *Starten*, um den Dienst zu starten.
 - **Den automatischen Start verhindern** Setzen Sie den Startmodus auf der Registerkarte *Dienste* auf *Manuell*.
3. Klicken Sie auf *OK*.

Sie können auch die Computerverwaltung verwenden, um Dienste zu konfigurieren. Um den automatischen Start eines Dienstes auf diese Weise einzurichten, gehen Sie wie folgt vor:

1. Klicken Sie auf die *Start*-Schaltfläche, zeigen Sie auf *Alle Programme* und *Verwaltung* und wählen Sie *Computerverwaltung*.
2. Standardmäßig werden Sie mit dem lokalen Computer verbunden. Um Verbindung mit einem Remote-computer aufzunehmen, rechtsklicken Sie auf den Knoten *Computerverwaltung* und wählen *Verbindung mit anderem Computer herstellen*. Wählen Sie im Dialogfeld *Computer auswählen* die Option *Anderer Computer* und geben Sie den Namen des Rechners ein. Dies kann in Form des Hostnamens geschehen, also z.B. als *EngDBSvr12*, oder als vollständig qualifizierter Domänenname wie *EngDBSvr12.cpandl.com*.
3. Erweitern Sie *Dienste und Anwendungen* und wählen Sie *Dienste*.
4. Rechtsklicken Sie auf den SQL Server-Dienst, den Sie automatisch starten lassen möchten, und wählen Sie *Eigenschaften*.
5. Sie können jetzt Folgendes tun:
 - **Den automatischen Start aktivieren** Setzen Sie auf der Registerkarte *Allgemein* den Starttyp auf *Automatisch*. Wenn als Status *Beendet* erscheint, klicken Sie auf *Starten*.
 - **Den automatischen Start verhindern** Setzen Sie den Starttyp auf der Registerkarte *Allgemein* auf *Manuell*.
6. Klicken Sie auf *OK*.

Festlegen der Startparameter für das Datenbankmodul

Die Startparameter steuern, wie das SQL Server-Datenbankmodul startet und welche Optionen dabei gesetzt werden. Die Startoptionen können Sie mit dem SQL Server-Konfigurations-Manager und mit der Computerverwaltung festlegen. Für diese Aufgabe wird der Konfigurations-Manager empfohlen, da er die aktuellen Standardeinstellungen anzeigt und es Ihnen ermöglicht, Änderungen auf einfache Weise vorzunehmen.

TIPP Startparameter können auch dem Befehlszeilenprogramm *SQLServr.exe* übergeben werden. Wenn Sie die Option *-c* übergeben, startet SQL Server ohne Verwendung eines Dienstes. Sie müssen *SQLServr.exe* aus dem Binn-Verzeichnis heraus starten, das zu der gewünschten Instanz des SQL Server-Datenbankmoduls gehört. Das Hilfsprogramm für die Standardinstanz befindet sich in *MSSQL11.MSSQLSERVER\MSSQL\Binn*. Bei benannten Instanzen lautet der Pfad *MSSQL11\$Instanzname\MSSQL\Binn*.

Hinzufügen von Startparametern

Sie können Startparameter wie folgt hinzufügen:

1. Starten Sie den SQL Server-Konfigurations-Manager mit einer der folgenden Methoden:
 - Melden Sie sich lokal oder über das Netzwerk am Datenbankserver an und starten Sie den SQL Server-Konfigurations-Manager. Im Menü *Microsoft SQL Server 2012* finden Sie die entsprechende Option unter *Konfigurationstools*.
 - Öffnen Sie in SQL Server Management Studio die Ansicht *Registrierte Server*, indem Sie **[STRG] + [ALT] + [G]** drücken. Wählen Sie über die Symbolleiste *Registrierte Server* die Gruppe der obersten Ebene aus und erweitern Sie die Gruppenknoten, indem Sie darauf doppelklicken. Rechtsklicken Sie auf den Servereintrag und wählen Sie *SQL Server-Konfigurations-Manager*.
2. Markieren Sie den Knoten *SQL Server-Dienste*. Rechtsklicken Sie auf den SQL Server-Dienst, den Sie automatisch starten lassen möchten, und wählen Sie *Eigenschaften*.
3. Klicken Sie auf der Registerkarte *Erweitert* in das Feld *Startparameter* und drücken Sie die Taste **[Ende]**, um zum Ende der bereits vorhandenen Parameter zu gelangen. Die Parameter *-d*, *-e* und *-l* sind standardmäßig gesetzt. Ändern Sie diese oder andere bereits vorhandene Parameter nicht versehentlich.
4. Die Parameter sind jeweils durch ein Semikolon voneinander getrennt. Geben Sie ein Semikolon und einen Bindestrich ein, gefolgt von dem Buchstaben und dem Wert des Parameters, den Sie hinzufügen möchten, z.B. *;-g512*.
5. Wiederholen Sie Schritt 3 und 4 für alle weiteren Parameter und Werte.
6. Klicken Sie auf *Übernehmen*, um die Änderungen zu speichern. Die Parameter werden beim nächsten Start der SQL Server-Instanz angewendet. Damit die Parameter sofort gültig werden, müssen Sie den Dienst beenden und dann wieder starten, indem Sie auf der Registerkarte *Anmelden* auf *Neu starten* klicken.

Entfernen von Startparametern

Sie können Startparameter wie folgt entfernen:

1. Starten Sie den SQL Server-Konfigurations-Manager mit einer der folgenden Methoden:
 - Melden Sie sich lokal oder über das Netzwerk am Datenbankserver an und starten Sie den SQL Server-Konfigurations-Manager, indem Sie auf die *Start-Schaltfläche* klicken, auf *Alle Programme*, *Microsoft SQL Server 2012* und *Konfigurationstools* zeigen und schließlich *SQL Server-Konfigurations-Manager* auswählen.

- Öffnen Sie in SQL Server Management Studio die Ansicht *Registrierte Server*, indem Sie **[Strg]** + **[ALT]** + **[G]** drücken. Wählen Sie über die Symbolleiste *Registrierte Server* die Gruppe der obersten Ebene aus und erweitern Sie die Gruppenknoten, indem Sie darauf doppelklicken. Rechtsklicken Sie auf den Servereintrag und wählen Sie *SQL Server-Konfigurations-Manager*.
2. Markieren Sie den Knoten *SQL Server-Dienste*. Rechtsklicken Sie auf den SQL Server-Dienst, den Sie automatisch starten lassen möchten, und wählen Sie *Eigenschaften*.
 3. Klicken Sie auf der Registerkarte *Erweitert* in das Feld *Startparameter*. Jeder Parameter wird mit einem Bindestrich, einem Buchstaben und einem Wert angegeben. Ein Semikolon dient zu Trennung der Parameterwerte, wie das folgende Beispiel zeigt: `-g512`;
 4. Entfernen Sie den Parameter, indem Sie seinen Eintrag löschen.
 5. Die Änderungen werden beim nächsten Start der SQL Server-Instanz angewendet. Damit sie sofort gültig werden, müssen Sie den Dienst beenden und dann wieder starten, indem Sie auf *Neu starten* klicken.

Übliche Startparameter

Tabelle 1.1 zeigt die Startparameter von SQL Server und ihre Verwendung. Die ersten drei (*-d*, *-e* und *-l*) sind die Standardparameter für SQL Server, mit den übrigen können Sie zusätzliche Einstellungen festlegen.

Parameter	Beschreibung
<code>-d<Pfad></code>	Legt den vollständigen Pfad für die <i>master</i> -Datenbank fest. Fehlt dieser Parameter, werden die Registrierungswerte verwendet. Beispiel: <code>-dC:\Programme\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\master.mdf</code>
<code>-e<Pfad></code>	Legt den vollständigen Pfad für das Fehlerprotokoll fest. Fehlt dieser Parameter, werden die Registrierungswerte verwendet. Beispiel: <code>-eC:\Programme\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\LOG\ERRORLOG</code>
<code>-l<Pfad></code>	Legt den vollständigen Pfad für das Transaktionsprotokoll der <i>master</i> -Datenbank fest. Fehlt dieser Parameter, werden die Registrierungswerte verwendet. Beispiel: <code>-lC:\Programme\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\mastlog.ldf</code>
<code>-B</code>	Legt einen Haltepunkt bei Fehlern fest; wird beim Debugging zusammen mit der Option <code>-y</code> verwendet.
<code>-c</code>	Verhindert, dass SQL Server als Dienst läuft. Das beschleunigt den Start, wenn SQL Server von der Befehlszeile aus ausgeführt wird.
<code>-E</code>	Erhöht die Anzahl der Blöcke, die den einzelnen Dateien in einer Dateigruppe zugewiesen sind. Dies ist für Data-Warehouse-Anwendungen mit einer begrenzten Anzahl von Benutzern nützlich.
<code>-f</code>	Startet SQL Server mit der Minimalkonfiguration. Diese Einstellung ist hilfreich, wenn ein Konfigurationswert den Start von SQL Server verhindert.
<code>-g<Zahl></code>	Gibt die Menge des virtuellen Adressspeichers in Megabyte an, die SQL Server reservieren soll. Dieser Speicher liegt außerhalb des Speicherpools und wird von DLLs für erweiterte Prozeduren, von OLE DB-Providern in verteilten Abfragen und dem Automationsobjekt in Transact-SQL verwendet. Der Standardwert beträgt 256. Beispiel: <code>-g256</code>
<code>-K</code>	Erzwingt die erneute Generierung des Diensthauptschlüssels, sofern einer vorhanden ist.
<code>-k<Zahl></code>	Setzt die Prüfpunktgeschwindigkeit in MB pro Sekunde (angegeben als dezimaler Wert). Beispiel: <code>-k25</code>

Tabelle 1.1 Startparameter für SQL Server