Microsoft

EXAM 70-462

Administering Microsoft® SQL Server® 2012 Databases

Orin Thomas
Peter Ward
boB Taylor

Training Kit

Exam 70-462: Administering Microsoft SQL Server 2012 Databases

OBJECTIVE	CHAPTER	LESSON
INSTALL AND CONFIGURE (19 PERCENT)		
Plan installation.	1	1
Install SQL Server and related services.	1	2
Implement a migration strategy.	4	1
Configure additional SQL Server components.	3	1
Manage SQL Server Agent.	11	1
MAINTAIN INSTANCES AND DATABASES (17 PERCENT)		
Manage and configure databases.	3	2
Configure SQL Server instances.	2	1
Implement a SQL Server clustered instance.	8	1
Manage SQL Server instances.	2	2
OPTIMIZE AND TROUBLESHOOT (14 PERCENT)		
Identify and resolve concurrency problems.	10	2
Collect and analyze troubleshooting data.	9	1–6
Audit SQL Server instances.	6	3
MANAGE DATA (20 PERCENT)		
Configure and maintain a back up strategy.	11	2
Restore databases.	11	3
Implement and maintain indexes.	10	1
Import and export data.	4	2
IMPLEMENT SECURITY (18 PERCENT)		
Manage logins and server roles.	5	1
Manage database permissions.	6	1
Manage users and database roles.	5	2
Troubleshoot security.	6	2
IMPLEMENT HIGH AVAILABILITY (12 PERCENT)		
Implement AlwaysOn.	8	2
Implement database mirroring.	7	1
Implement replication.	7	2

Exam Objectives The exam objectives listed here are current as of this book's publication date. Exam objectives are subject to change at any time without prior notice and at Microsoft's sole discretion. Please visit the Microsoft Learning website for the most current listing of exam objectives: *http://www.microsoft.com/learning/en/us/exams/70-462.mspx.*

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2012938612
ISBN: 978-0-7356-6607-8

Printed and bound in the United States of America.

Fourth Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at http://www.microsoft.com/learning/booksurvey.

Acquisitions Editor: Anne Hamilton
Developmental Editor: Karen Szall
Project Editor: Karen Szall
Editorial Production: nSight, Inc.
Technical Reviewer: boB Taylor; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Copyeditor: Kerin Forsyth
Indexer: Lucie Haskins
Cover: Twist Creative • Seattle

[QG] [2013-05-24]

Contents

Contents

Introduction

This training kit is designed for information technology (IT) professionals who support or plan to support Microsoft SQL Server 2012 databases and who also plan to take Exam 70-462, "Administering Microsoft SQL Server 2012 Databases." It is assumed that before you begin using this kit, you have a solid, foundation-level understanding of SQL Server 2012 and have used the product extensively either in one of the release candidate versions or with the release to manufacturing (RTM) version. Although this book helps prepare you for the 70-462 exam, you should consider it one part of your exam preparation plan. You require meaningful, real-world experience with SQL Server 2012 to pass this exam.

The material covered in this training kit and on exam 70-462 relates to the technologies in SQL Server 2012. The topics in this training kit cover what you need to know for the exam as described on the Skills Measured tab for the exam, which is available at *http://www.microsoft.com/learning/en/us/exam.aspx?ID=70-462&locale=en-us#tab2*.

By using this training kit, you will learn how to do the following:

- Install and configure SQL Server 2012
- Manage SQL Server instances and databases
- Optimize and troubleshoot SQL Server 2012
- Manage SQL Server 2012 data
- Implement instance and database security
- Implement high availability

Refer to the objective mapping page in the front of this book to see where in the book each exam objective is covered.

System Requirements

The following are the minimum system requirements your computer needs to meet to complete the practice exercises in this book and to run the companion CD. To minimize the time and expense of configuring physical computers for this training kit, it's recommended that you use Hyper-V, which is a feature of Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and certain editions of Windows 8. You can use other virtualization software instead, but the instructions are written assuming that you are using a solution that supports 64-bit operating systems hosted as virtual machines.

Hardware Requirements

This section presents the hardware requirements for Hyper-V, the hardware requirements if you are not using virtualization software, and the software requirements.

Virtualization Hardware Requirements

If you choose to use virtualization software, you need only one physical computer to perform the exercises in this book. That physical host computer must meet the following minimum hardware requirements:

- x64-based processor that includes both hardware-assisted virtualization (AMD-V or Intel VT) and hardware data execution protection. (On AMD systems, the data execution protection feature is called the No Execute or NX bit. On Intel systems, this feature is called the Execute Disable or XD bit.) These features must also be enabled in the BIOS. (Note: You can run Windows Virtual PC without Intel-VT or AMD-V.)
- 8.0 GB of RAM.
- 80 GB of available hard disk space if you are using differencing virtual hard disks.
- DVD-ROM drive.
- Internet connectivity.

Physical Hardware Requirements

If you choose to use physical computers instead of virtualization software, use the following list to meet the minimum hardware requirements of the practice exercises in this book:

- Six personal computers, each with a 1.4-GHz, 64-bit processor, minimum 2 GB of RAM, 50 GB hard disk drive, network card, video card, and DVD-ROM drive.
- All six computers must be connected to the same network.

Software Requirements

The following software is required to complete the practice exercises:

- **Windows Server 2008 R2** You can download an evaluation edition of Windows Server 2008 R2 at the Microsoft Download Center at *http://www.microsoft.com/downloads*.
- **SQL Server 2012** You can download an evaluation edition of SQL Server 2012 at the Microsoft Download Center at *http://www.microsoft.com/downloads*.
- **AdventureWorks2012 and AdventureWorksDW2012 databases** These can be obtained through this book's companion content page at *http://go.microsoft.com/FWLink/?Linkid=251256*.

Practice Setup Instructions

This section contains abbreviated instructions for setting up the domain controller (DC), SQL-A, SQL-B, SQL-C, SQL-D, and SQL-Core computers used in the practice exercises in all chapters of this training kit. To perform these exercises, first install Windows Server 2008 R2 Enterprise edition with Service Pack 1 using the default configuration, setting the administrator password to **Pa$$w0rd**. For server SQL-Core, install Windows Server 2008 R2 Enterprise Edition with Service Pack 1 in the default server core configuration, setting the administrator password to **Pa$$w0rd**.

IMPORTANT **DOWNLOAD REQUIRED SOFTWARE**

Before you begin preparing the practice computers, you must have a copy of Windows Server 2008 R2 Enterprise edition with Service Pack 1 (either as an .iso file or as a DVD).

Prepare a Computer to Function as a Windows Server 2008 R2 Domain Controller

1. Log on to the first computer on which you have installed Windows Server 2008 R2 with Service Pack 1, using the Administrator account and the password **Pa$$w0rd**.
2. Open an elevated command prompt and issue the following commands:

   ```
   Netsh interface ipv4 set address "Local Area Connection" static 10.10.10.10
   ```

3. Enter the following command:

   ```
   netdom renamecomputer %computername% /newname:DC
   ```

4. Restart the computer and log on again, using the Administrator account.
5. Click Start. In the Search Programs And Files text box, type the following:

   ```
   Dcpromo.
   ```

6. When the Active Directory Domain Services Installation Wizard starts, click Next twice.
7. On the Choose A Deployment Configuration page, choose Create A New Domain In A New Forest and then click Next.
8. On the Name The Forest Root Domain page, enter Contoso.com, and then click Next.
9. On the Forest Functional Level page, set the forest functional level to Windows Server 2008 R2 and then click Next.
10. On the Set Domain Functional Level page, ensure that Windows Server 2008 R2 is set and then click Next.

11. On the Additional Domain Controller Options page, ensure that the DNS Server option is selected and then click Next. When presented with the warning that the delegation for the DNS server cannot be created, click Yes when asked whether you want to continue.
12. Accept the default settings for the Database, Log Files, and SYSVOL locations and click Next.
13. In the Directory Services Restore Mode Administrator Password dialog box, enter the password **Pa$$w0rd** twice, and then click Next.
14. On the Summary page, click Next to begin the installation of Active Directory Domain Services (AD DS) on computer DC. When the wizard completes, click Finish. When prompted, click Restart Now to reboot computer DC.

Prepare AD DS

1. Log on to server DC, using the Administrator account.
2. Using Active Directory Users And Computers, create a user account named Kim_Akers in the Users container and assign the account the password **Pa$$w0rd**. Configure the password to never expire. Add this user account to the Enterprise Admins, Domain Admins, and Schema Admins groups.
3. Open the DNS console and create a primary reverse lookup zone for the subnet 10.10.10.x. Ensure that the zone is stored within AD DS and is replicated to all DNS servers running on domain controllers in the forest.

Prepare a Member Server and Join It to the Domain

1. Ensure that computer DC is turned on and connected to the network or virtual network to which the second computer is connected.
2. Log on to the second computer on which you have installed Windows Server 2008 R2 with Service Pack 1, using the Administrator account and the password **Pa$$w0rd**.
3. Open an elevated command prompt and issue the following commands:

```
Netsh interface ipv4 set address "Local Area Connection" static 10.10.10.20

Netsh interface ipv4 set dnsservers "Local Area Connection" static 10.10.10.10
primary
```

4. Enter the following command:

```
netdom renamecomputer %computername% /newname:SQL-A
```

5. Restart the computer and then log on again, using the Administrator account.

6. From an elevated command prompt, issue the following command:

```
netdom join SQL-A /domain:contoso.com
```

7. Restart the computer. When the computer restarts, log on as contoso\Administrator and then turn off the computer.

Prepare a Second Member Server and Join It to the Domain

1. Ensure that computer DC is turned on and connected to the network or virtual network to which the second computer is connected.
2. Log on to the third computer on which you have installed Windows Server 2008 R2 with Service Pack 1, using the Administrator account and the password **Pa$$w0rd**.
3. Open an elevated command prompt and issue the following commands:

```
Netsh interface ipv4 set address "Local Area Connection" static 10.10.10.30

Netsh interface ipv4 set dnsservers "Local Area Connection" static 10.10.10.10
primary
```

4. Enter the following command:

```
netdom renamecomputer %computername% /newname:SQL-B
```

5. Restart the computer and then log on again, using the Administrator account.
6. From an elevated command prompt, issue the following command:

```
netdom join SQL-B /domain:contoso.com
```

7. Restart the computer. When the computer restarts, log on as contoso\Administrator. Turn off the computer.

Prepare a Third Member Server and Join It to the Domain

1. Ensure that computer DC is turned on and connected to the network or virtual network to which the second computer is connected.
2. Log on to the third computer that you have installed Windows Server 2008 R2 with Service Pack 1 on using the Administrator account and the password **Pa$$w0rd**.
3. Open an elevated command prompt and issue the following commands:

```
Netsh interface ipv4 set address "Local Area Connection" static 10.10.10.40

Netsh interface ipv4 set dnsservers "Local Area Connection" static 10.10.10.10
primary
```

4. Enter the following command:

```
netdom renamecomputer %computername% /newname:SQL-C
```

5. Restart the computer and then log on again using the Administrator account.
6. From an elevated command prompt, issue the following command:

```
netdom join SQL-C /domain:contoso.com
```

7. Restart the computer. When the computer restarts, log on as contoso\Administrator. Turn off the computer.

Prepare a Fourth Member Server and Join It to the Domain

1. Ensure that computer DC is turned on and connected to the network or virtual network to which the second computer is connected.
2. Log on to the fourth computer on which you have installed Windows Server 2008 R2 with Service Pack 1, using the Administrator account and the password **Pa$$w0rd**.
3. Open an elevated command prompt and issue the following commands:

```
Netsh interface ipv4 set address "Local Area Connection" static 10.10.10.50

Netsh interface ipv4 set dnsservers "Local Area Connection" static 10.10.10.10
primary
```

4. Enter the following command:

```
netdom renamecomputer %computername% /newname:SQL-D
```

5. Restart the computer and then log on again, using the Administrator account.
6. From an elevated command prompt, issue the following command:

```
netdom join SQL-D /domain:contoso.com
```

7. Restart the computer. When the computer restarts, log on as contoso\Administrator. Turn off the computer.

Prepare a Computer Running the Server Core Installation Option and Join It to the Domain

1. Ensure that computer DC is turned on and connected to the network or virtual network to which the second computer is connected.
2. Using the Administrator account and the password **Pa$$w0rd,** log on to the computer on which you have installed Windows Server 2008 R2 with Service Pack 1 in the Server Core configuration.

3. From the Administrator command prompt, enter the following commands:

```
Netsh interface ipv4 set address "Local Area Connection" static 10.10.10.60

Netsh interface ipv4 set dnsservers "Local Area Connection" static 10.10.10.10
primary
```

4. Enter the following command to configure the computer's name:

```
netdom renamecomputer %computername% /newname:SQL-CORE
```

5. Enter the following command to restart the computer:

```
Shutdown /r /t 5
```

6. Restart the computer and log back on, using the Administrator account.
7. Enter the following command to join the computer to the domain:

```
netdom join SQL-CORE /domain:contoso.com
```

8. Enter the following command to restart the computer:

```
Shutdown /r /t 5
```

9. Restart the computer. When the computer restarts, log on as contoso\Administrator. Turn off the computer, using the following command:

```
Shutdown /s /t 5
```

Using the Companion CD

A companion CD is included with this training kit. The companion CD contains the following:

- **Practice tests** You can reinforce your understanding of the topics covered in this training kit by using electronic practice tests that you customize to meet your needs. You can practice for the 70-462 certification exam by using tests created from a pool of 200 practice exam questions, which give you many practice exams to help you prepare for the certification exam. These questions are not from the exam; they are for practice and preparation.
- **An eBook** An electronic version (eBook) of this book is included for when you do not want to carry the printed book with you.

> ***NOTE*** **SAMPLE SQL SERVER 2012 DATABASES**
>
> **The practices in this book rely on two sample databases: AdventureWorks2012 and AdventureWorksDW2012. You can download these databases for your use from the book's companion content page at *http://go.microsoft.com/FWLink/?Linkid=251256*.**

How to Install the Practice Tests

To install the practice test software from the companion CD to your hard disk, perform the following steps:

1. Insert the companion CD into your CD drive and accept the license agreement. A CD menu appears.

> ***NOTE*** **IF THE CD MENU DOES NOT APPEAR**
>
> **If the CD menu or the license agreement does not appear, AutoRun might be disabled on your computer. Refer to the Readme.txt file on the CD for alternate installation instructions.**

2. Click Practice Tests and follow the instructions on the screen.

How to Use the Practice Tests

To start the practice test software, follow these steps:

1. Click Start, All Programs, and then select Microsoft Press Training Kit Exam Prep.

   A window appears that shows all the Microsoft Press training kit exam prep suites installed on your computer.

2. Double-click the practice test you want to use.

When you start a practice test, you choose whether to take the test in Certification Mode, Study Mode, or Custom Mode:

- **Certification Mode** Closely resembles the experience of taking a certification exam. The test has a set number of questions. It is timed, and you cannot pause and restart the timer.
- **Study Mode** Creates an untimed test during which you can review the correct answers and the explanations after you answer each question.
- **Custom Mode** Gives you full control over the test options so that you can customize them as you like.

In all modes, the user interface when you are taking the test is basically the same but with different options enabled or disabled, depending on the mode.

When you review your answer to an individual practice test question, a "References" section is provided that lists where in the training kit you can find the information that relates to that question and provides links to other sources of information. After you click Test Results to score your entire practice test, you can click the Learning Plan tab to see a list of references for every objective.

How to Uninstall the Practice Tests

To uninstall the practice test software for a training kit, use the Program And Features option in Windows Control Panel.

Acknowledgments

A book is put together by many more people than the authors whose names are listed on the cover page. We'd like to express our gratitude to the following people for all the work they have done in getting this book into your hands: Karen Szall, boB Taylor, Carol Whitney, Kerin Forsyth, and Lucie Haskins.

Errata & Book Support

We've made every effort to ensure the accuracy of this book and its companion content. Any errors that have been reported since this book was published are listed on our Microsoft Press site:

http://go.microsoft.com/FWLink/?Linkid=251255

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software is not offered through the addresses above.

We Want to Hear from You

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://www.microsoft.com/learning/booksurvey

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in Touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

Preparing for the Exam

Microsoft certification exams are a great way to build your resume and let the world know about your level of expertise. Certification exams validate your on-the-job experience and product knowledge. While there is no substitution for on-the-job experience, preparation through study and hands-on practice can help you prepare for the exam. We recommend that you round out your exam preparation plan by using a combination of available study materials and courses. For example, you might use the Training Kit and another study guide for your "at home" preparation, and take a Microsoft Official Curriculum course for the classroom experience. Choose the combination that you think works best for you.

NOTE **PASSING THE EXAM**

Take a minute (well, one minute and two seconds) to look at the "Passing a Microsoft Exam" video at *http://www.youtube.com/watch?v=Jp5qg2NhgZ0&feature=youtu.be*. It's true. Really!

CHAPTER 3

Configuring SQL Server 2012 Components

Exam objectives in this chapter:

- Configure additional SQL Server components.
- Manage and configure databases.

Microsoft SQL Server 2012 database administrators (DBAs) must be able to deploy not only the Database Engine but also Analysis Services and Reporting Services in either Native or SharePoint integrated mode. In this chapter, you learn about the new FileTable feature and configuring Integration Services security and support for FILESTREAM. As a DBA, you must also know how to create full-text indexes, design filegroups, partition tables and indexes, and how to configure transparent data encryption and data compression.

Lessons in this chapter:

Before You Begin

To complete the practice exercises in this chapter, you must have:

- Completed the setup tasks for installing computers DC, SQL-A, SQL-B, and SQL-CORE as outlined in the introduction of this book.
- Completed the setup tasks outlined in the end-of-lesson practice exercises in Chapter 1, "Planning and Installing SQL Server 2012," and Chapter 2, "Configuring and Managing SQL Server Instances."

No additional configuration is required for this chapter.

Lesson 1: Configuring Additional SQL Server Components

SQL Server 2012 is more than just the Database Engine. In this lesson, you learn how to deploy SQL Server 2012 Analysis Services, Reporting Services, SharePoint integration, full-text indexing, and SQL Server Integration Services and how to configure the FILESTREAM and FileTable features.

After this lesson, you will be able to:

- Deploy and configure Analysis Services.
- Deploy and configure Reporting Services.
- Deploy and configure SharePoint integration.
- Manage full-text indexing.
- Configure SQL Server Integration Services security.
- Configure FILESTREAM and FileTable.

Estimated lesson time: 60 minutes

Deploying and Configuring Analysis Services

When installing Analysis Services, you can choose whether to install it in *multidimensional mode* and *data mining mode* or in *tabular mode*, as shown in Figure 3-1. Mode is specific to an instance, and if you want to use more than one mode, it is necessary to install more than one Analysis Services instance. The difference between these modes is as follows:

- **Multidimensional and data mining mode** The default Analysis Services mode. Supports online analytical processing (OLAP) databases and data mining models.
- **Tabular mode** Supports new tabular modeling features. When installed using this mode, Analysis Services can host solutions built in the tabular model designer. Analysis Services in tabular mode is necessary when you want tabular model data access over a network.

You can install Analysis Services from the command line by using the /FEATURES=AS option. The /ASSERVERMODE can be set to MULTIDIMENSIONAL, TABULAR, or POWERPIVOT. For example, to create an instance named ASMulti with Analysis Services installed in multi-dimensional and data mining mode and configuring the Analysis Services service account as contoso\asaccount, and with contoso\kim_akers as the Analysis Services Administrator account, use the following command:

```
Setup.exe /q /IAcceptSQLServerLicenseTerms /Action=install /Features=AS
/ASSERVERMODE=MULTIDIMENSIONAL /INSTANCENAME=ASMulti /ASSVCACCOUNT=Contoso\kim_akers
/ASSYSADMINACCOUNTS=contoso\kim_akers
```

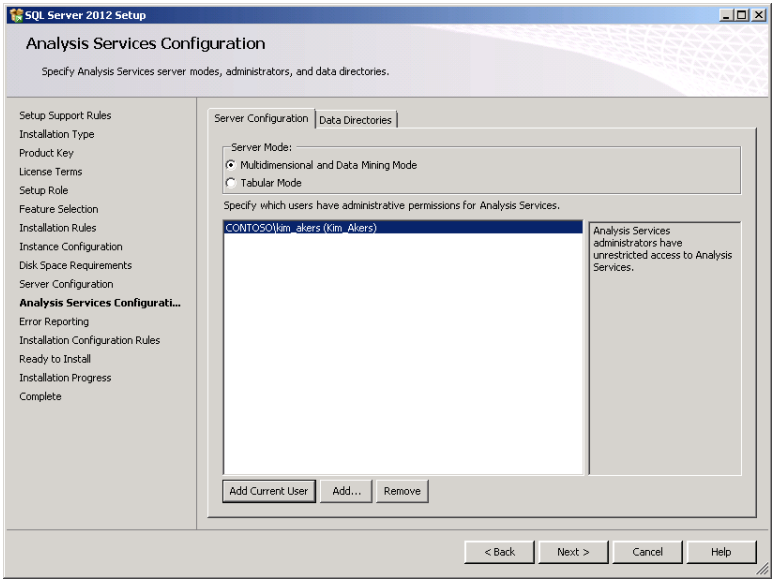

FIGURE 3-1 Multidimensional and data mining mode or tabular mode

To create an instance named ASTabular with Analysis Services installed in tabular mode, with the Analysis Services service account as NetworkService, and with contoso\kim_akers as the Analysis Services Administrator account, use the following command:

```
Setup.exe /q /IAcceptSQLServerLicenseTerms /Action=install /Features=AS
/ASSERVERMODE=TABULAR /INSTANCENAME=ASTabular /ASSVCACCOUNT=NetworkService
/ASSYSADMINACCOUNTS=contoso\kim_akers
```

Although the option isn't available in the dialog box displayed in Figure 3-1, you can also deploy Analysis Services in PowerPivot for SharePoint mode by using the /ASSSERVERMODE=POWERPIVOT installation option.

> ***MORE INFO*** **INSTALLING ANALYSIS SERVICES**
>
> **You can learn more about installing Analysis Services at *http://msdn.microsoft.com/en-us/library/hh231722(SQL.110).aspx*.**

Analysis Services uses a managed service account when installed by default. You can also configure Analysis Services to use a domain or local user account:

- If Analysis Services will connect to network resources in the security context of the logon account, create a specific domain user account for use with Analysis Services. You can also use the NetworkService account. When you use this account, the account presents the local computer's credentials to remote servers. To grant access to this account, use the Computer account of the Analysis Server host.

- If Analysis Services will not connect to external network resources, Analysis Services can be run using a local user account, a domain user account, a virtual account, or a managed service account.

Best practice is to run Analysis Services by using an account assigned the fewest possible privileges. You should avoid using the LocalService and NetworkService accounts in high-security environments because Analysis Services connection strings and passwords can be decrypted and are accessible to the Analysis Services logon account.

> ***MORE INFO*** **ANALYSIS SERVICES ACCOUNTS**
>
> **You can learn more about Analysis Services accounts at *http://msdn.microsoft.com/en-us/library/ms174905(SQL.110).aspx*.**

Deploying and Configuring Reporting Services

To install a SQL Server Reporting Services (SSRS) Native Mode Report Server–only instance by using SQL Server Installation Center, perform the following general steps:

1. Open SQL Server Installation Center from the Configuration Tools folder.
2. Click Installation and then choose New SQL Server Stand-Alone Installation Or Add Features To An Existing Installation. Specify the location of the SQL Server 2012 installation files.
3. Click OK after the Setup Support Rules check runs.
4. Click Next on the Product Updates page.
5. Click Next on the Setup Support Rule page.
6. On the Installation Type page, choose Perform A New Installation Of SQL Server 2012.
7. On the Product Key page, enter the product key.
8. On the License Terms page, choose I Accept The License Terms.
9. On the Setup Role page, choose SQL Server Feature Installation.
10. On the Feature Selection page, choose Reporting Services - Native And Database Engine Services shown in Figure 3-2, and then click Next.

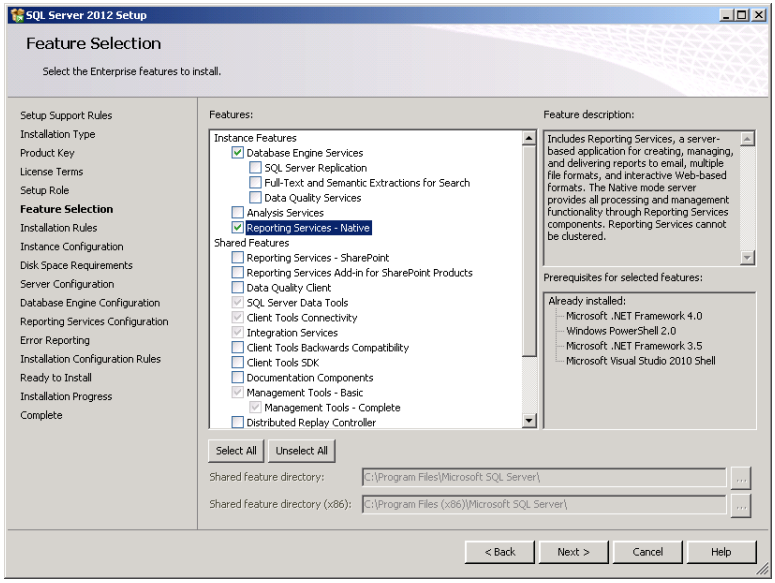

FIGURE 3-2 Install Reporting Services

11. On the Installation Rules page, click Next.
12. On the Instance Configuration page, provide a name for the Reporting Services instance.
13. On the Disk Space Requirements page, click Next.
14. On the Service Accounts page, review the Service Account configuration and then click Next.
15. On the Database Engine Configuration page, add the users who will hold the SQL Server Administrative role and then click Next.
16. On the Reporting Services Configuration page, shown in Figure 3-3, choose Install And Configure. You'll have this option only if you have already installed the necessary Web Server components.

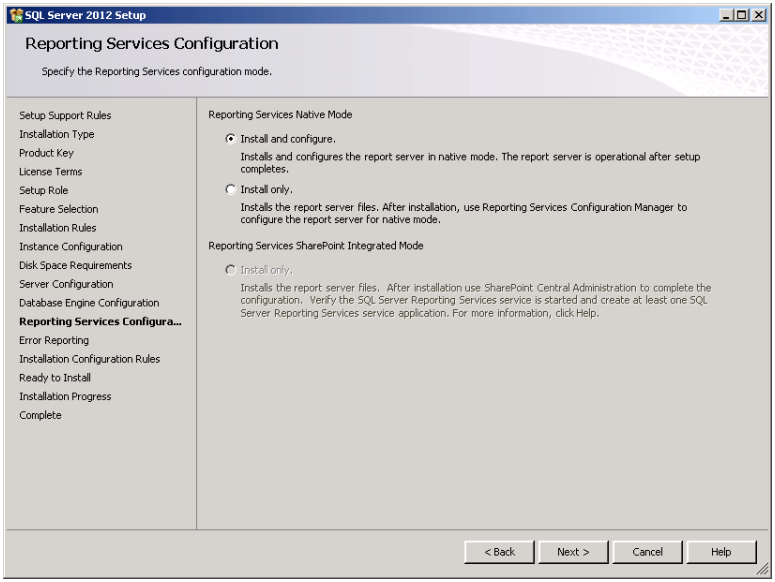

FIGURE 3-3 Install and configure Reporting Services

17. On the Error Reporting page, click Next twice and then choose Install. Click Close to dismiss the Setup Wizard.

EXAM TIP

You can choose the Install And Configure option only if you have installed the web server role prior to attempting to install the Report Server instance and you are also installing the Database Engine on the same instance.

To install Reporting Services in the default configuration for native mode from the command line from the command prompt, by using the NetworkService account for both the Reporting Services service account and the SQL Server service account and assigning members of BUILTIN\ADMINISTRATORS SQL Server system administrator access, and by using the instance named RPTSVR, use the following command:

```
setup /q /IAcceptSQLServerLicenseTerms /ACTION=install /FEATURES=SQL,RS,TOOLS
/INSTANCENAME=RPTSVR /SQLSYSADMINACCOUNTS="BUILTIN\ADMINISTRATORS"
/RSSVCACCOUNT=NetworkService /SQLSVCACCOUNT=NetworkService /AGTSVCACCOUNT=NetworkService
/RSSVCSTARTUPTYPE="Manual" /RSINSTALLMODE="DefaultNativeMode"
```

MORE INFO **REPORTING SERVICES**

You can learn more about installing Reporting Services at *http://msdn.microsoft.com/en-us/library/ms143711(SQL.110).aspx*.

If you install Reporting Services Configuration Manager by using the Install And Configure option, Reporting Services is already configured for you. If you install Reporting Services by using the files-only installation option, you must run Reporting Services Configuration Manager, shown in Figure 3-4, to configure Reporting Services.

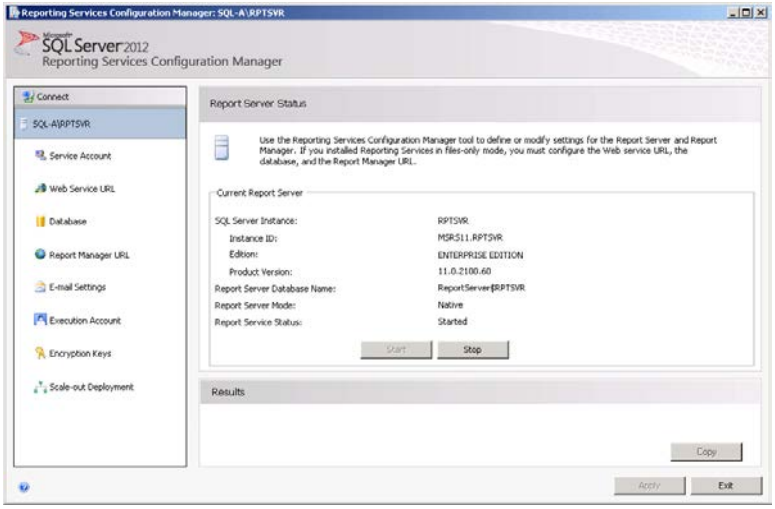

FIGURE 3-4 Reporting Services Configuration Manager

Using Reporting Services Configuration Manager, which you can launch from the Configuration menu, you can perform the following tasks:

- Configure the Reporting Services service account.
- Configure the Web Service URL, including Virtual directory, IP Address, TCP Port, SSL Certificate, and SSL Port.
- Configure the Report Server Database and database credential.
- Configure the Report Manager URL.
- Configure email settings, including Sender Address, Current SMTP Delivery Method, and SMTP Server.
- Configure the Execution Account. This is usually a domain account with minimal permissions that is used for retrieving external report data sources that do not require authentication and for unattended report processing.
- Configure Backup And Restore and update Reporting Services encryption keys.
- Configure Scale-out Deployment.

MORE INFO **REPORTING SERVICES CONFIGURATION**

You can learn more about Reporting Services Configuration at *http://msdn.microsoft.com/en-us/library/ms157412(SQL.110).aspx*.

Deploying and Configuring SharePoint Integration

You can deploy Analysis Services and Reporting Services as shared services in a SharePoint farm. Deploying Analysis Services and Reporting Services enables you to use features such as PowerPivot for Microsoft SharePoint and Power View, a Reporting Services interactive report designer.

To deploy Reporting Services, Power View, and PowerPivot for SharePoint, you must install the following products:

- SharePoint Server 2010 Enterprise edition with Service Pack 1
- SQL Server 2012 Database Engine
- SQL Server 2012 Reporting Services and Reporting Services Add-in
- SQL Server 2012 PowerPivot for SharePoint

The host computer must be joined to the domain, and you must configure domain user accounts for the following services:

- SharePoint Web Services and Administrative Services
- Reporting Services
- Analysis Services
- Microsoft Excel Services
- Secure Store Services
- PowerPivot System Service

The SQL Server 2012 Database Engine can use a Virtual or Managed service account. To configure SharePoint 2010 and SQL Server 2012 integration, perform the following general steps:

1. Install a SharePoint Server 2010 SP1 Enterprise edition farm. Choose to configure the farm later by not running the SharePoint 2010 Product Configuration Wizard. This enables you to use the SQL Server 2012 Database Engine as the farm's database server.
2. Install the SQL Server 2012 Database Engine and PowerPivot for SharePoint, as shown in Figure 3-5.

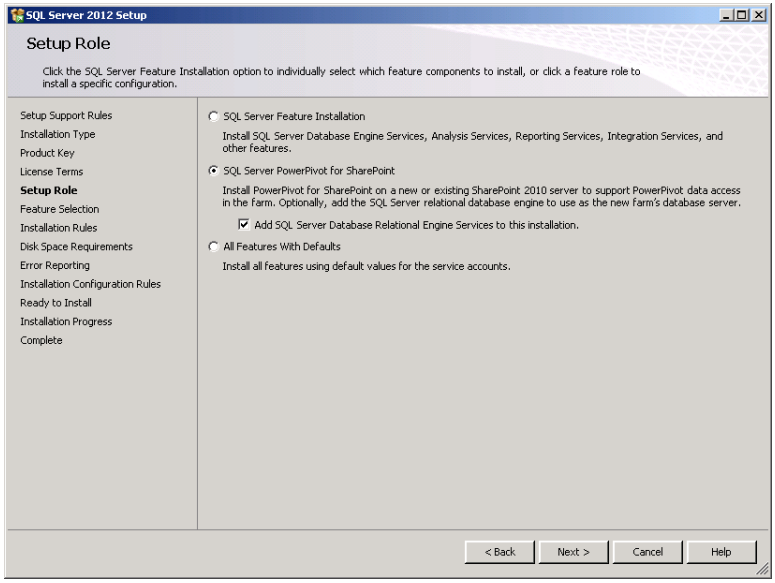

FIGURE 3-5 Install SQL Server PowerPivot for SharePoint

3. Accept the default instance ID of PowerPivot and complete the SQL Server 2012 Installation Wizard.
4. Use the PowerPivot Configuration tool, available from the Configuration Tools folder and shown in Figure 3-6, to create the farm, a default web application, and a root site collection.

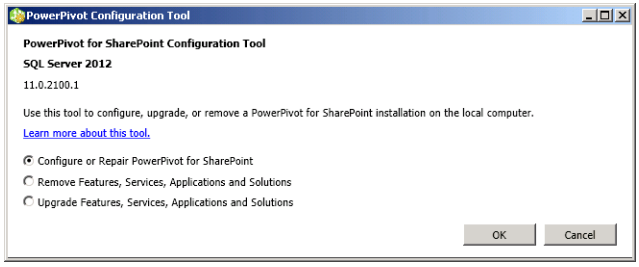

FIGURE 3-6 PowerPivot Configuration Tool

5. Verify that the farm is operational by navigating to Central Administration.
6. Run SQL Server 2012 setup again to install and configure Reporting Services and the Reporting Services Add-in.

7. SharePoint Site Administrators can extend SharePoint document libraries to use Business Intelligence (BI) content types. This can be done by performing the following steps:
   A. In Shared Documents or another document library, on the Library tab, click Library Settings. Under General Settings, click Advanced Settings. In Content Types, click Yes to enable management of content types.
   B. On the Library tab, click Library Settings. Under Content Types, click Add From Existing Site Content Types. Locate the Business Intelligence content type group and add BI Semantic Model Connection File and Report Data Source.
8. SharePoint Site Administrators create data connection files to launch Power View. This involves creating a BI semantic model connection (.bism) or a Reporting Services shared data source (.rsds) as a data source for Power View.

MORE INFO **SHAREPOINT INTEGRATION**

You can learn more about SharePoint 2010 integration with SQL Server 2012 at *http://msdn.microsoft.com/en-us/library/hh231671(SQL.110).aspx*.

Quick Check

- **Which mode should you select during the installation of Analysis Services if you want to support OLAP databases?**

Quick Check Answer

- **You should install Analysis Services in multidimensional and data mining mode if you want to use OLAP databases.**

Configuring SQL Server Integration Services Security

Integration Services enables you to run and schedule Integration Services packages in SQL Server Management Studio. You can install Integration Services only once on a computer, even if that computer hosts multiple instances. You can install Integration Services as a shared feature in the SQL Server Setup Wizard or install it from the command line by issuing the following command:

```
Setup.exe /q /IAcceptSQLServerLicenseTerms /Action=Install /Features=IS
```

In previous versions of SQL Server, all users in the Users group could access the Integration Services service. In SQL Server 2012, the service is secure by default; and, by default, only the built-in Administrators group can run Integration Services. You must use the DCOM

Configuration tool (dcomcnfg.exe) to grant specific users access to SQL Server Integration Services (SSIS). To do this, perform the following steps:

1. Run Dcomcnfg.exe from the Search Programs and Files text box.
2. Expand the Component Services, Computers, My Computer, and DCOM Config nodes.
3. Right-click Microsoft SQL Server Integration Services 11.0 and choose Properties.
4. On the Security tab, shown in Figure 3-7, click Edit in Launch And Activation Permissions.

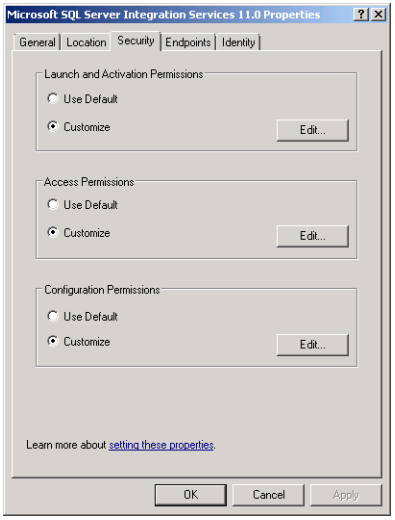

FIGURE 3-7 Use Dcomcnfg.exe for Integration Services permissions

5. Add users and assign permissions. You can assign the following permissions:
   - Local Launch
   - Remote Launch
   - Local Activation
   - Remote Activation

MORE INFO **INTEGRATION SERVICES**

You can learn more about Integration Services at *http://msdn.microsoft.com/en-us/library/ms143731(SQL.110).aspx*.

Managing Full-Text Indexing

Full-text indexes store information about significant words and their location within the columns of a database table. In SQL Server 2012, the Full-Text Engine is part of the SQL Server process rather than a separate service. Only one full-text index can be created per table or indexed view. A full-text index can contain up to 1,024 columns.

To create a full-text index by using SQL Server Management Studio, perform the following steps:

1. Right-click the table on which you want to create a new full-text index and choose Design.
2. From the Table Designer menu, click Full-Text Index to open the Full-Text Index dialog box. Click Add. Configure the properties of the index as shown in Figure 3-8.

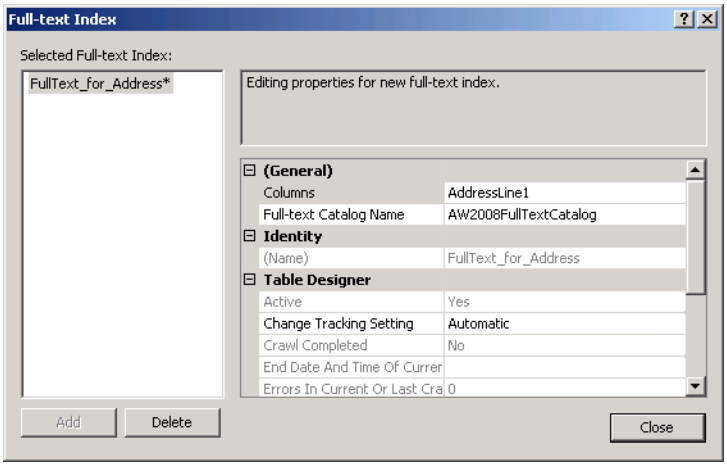

FIGURE 3-8 Full-text Index

As an alternative to using the Table Designer, you can run the Full-Text Indexing Wizard by performing the following steps:

1. Right-click the table for which you want to configure the full-text index, choosing Full-Text Index and then Define Full-Text Index. This launches the Full-Text Indexing Wizard.
2. On the Select An Index page, choose a unique index for the table.
3. On the Select Table Columns page, choose the columns you want to be eligible for full-text queries, as shown in Figure 3-9.

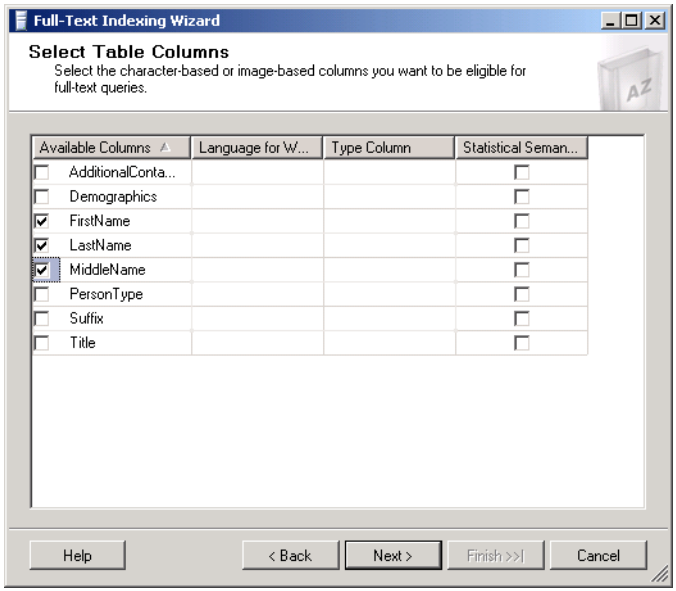

FIGURE 3-9 Full-text index columns

4. On the Select Change Tracking page, choose whether to track changes as they occur, to track them manually, or not to track changes.
5. On the Select Catalog, Index Filegroup, And Stoplist page, you can choose to use an existing full-text catalog or create a new catalog. You can also select the index filegroup and the full-text stoplist. Figure 3-10 shows the creation of a new catalog named NEWCATALOG.

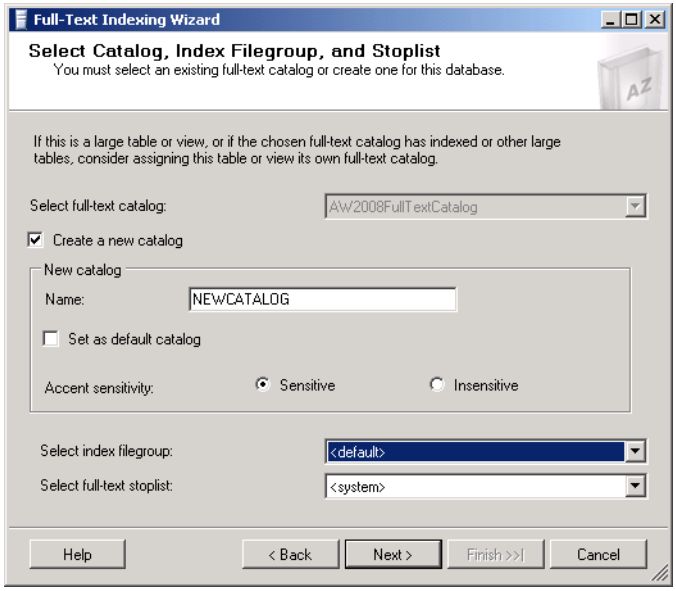

FIGURE 3-10 New full-text catalog

6. On the Define Population Schedules page, you can specify a population schedule for the full-text catalog.
7. On the Summary page, click Finish to complete creation of the new full-text index.

You can use the CREATE FULLTEXT INDEX statement to create a full-text index on a table. For example, to create a FULLTEXT index on the Production.ProductReview table in the AdventureWorks2012 database by using the ReviewerName, EmailAddress, and Comments columns in the existing unique key index PK_ProductReview_ProductReviewID while also creating a FULLTEXT catalog called production_catalog, use the following statement:

```
USE AdventureWorks2012;
GO
CREATE FULLTEXT CATALOG production_catalog;
GO
CREATE FULLTEXT INDEX ON Production.ProductReview
(
ReviewerName, EmailAddress, Comments
)
KEY INDEX PK_ProductReview_ProductReviewID
ON production_catalog;
GO
```

To delete a full-text index, right-click the table that hosts the full-text index, choose Full-Text Index, and then select Delete Full-Text Index. You can also delete a full-text index by using the DROP FULLTEXT INDEX statement. For example, to drop the index created in the previous example, use the query:

```
DROP FULLTEXT INDEX ON Production.ProductReview
```

MORE INFO **FULL-TEXT INDEXES**

You can learn more about full-text indexes at *http://msdn.microsoft.com/en-us/library/cc879306(v=SQL.110).aspx*.

Configuring FILESTREAM

FILESTREAM enables SQL Server–based applications to store unstructured data, such as images and documents, on the host computer's file system. To use FILESTREAM, you must create or modify a database to host a special type of filegroup, after which you can create or modify tables so that they can use the varbinary(max) column with the FILESTREAM attribute.

You should use FILESTREAM under the following conditions:

- Objects that you want to store are greater than 1 MB. The traditional varbinary(max) limit of 2 GB does not apply to BLOBs (binary large objects) stored in the file system.
- Fast read access is important.

For objects smaller than 1 MB, use the varbinary(max) BLOB data type. You can't enable FILESTREAM if you are running a 32-bit version of SQL Server 2012 on a 64-bit operating system.

To enable FILESTREAM, perform the following steps:

1. Open SQL Server Configuration Manager from the Configuration Tools folder.
2. Edit the properties of the instance on which you want to enable FILESTREAM.
3. On the FILESTREAM tab, select Enable FILESTREAM For Transact-SQL Access. You can also use this dialog box to enable FILESTREAM for file I/O streaming access and to allow remote clients access to FILESTREAM data, as shown in Figure 3-11.

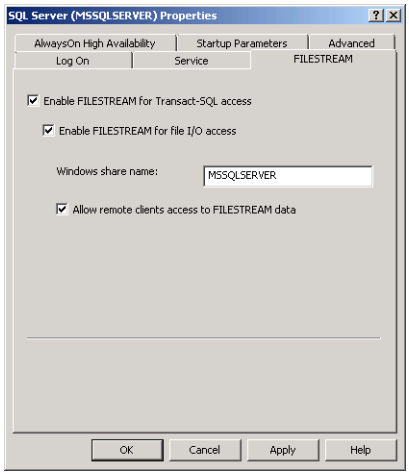

FIGURE 3-11 Enabling FILESTREAM

4. In SQL Server Management Studio, execute the following query:

```
EXEC sp_configure filestream_access_level, 2
RECONFIGURE
```

> ***NOTE*** **FILESTREAM_ACCESS_LEVEL**
>
> **Setting filestream_access_level to 0 disables FILESTREAM access. Setting level 1 allows Transact-SQL only. Setting level 2 allows Transact-SQL and Win32 streaming.**

5. Restart the SQL Server Service related to the instance on which you are enabling FILESTREAM by using SQL Server Configuration Manager.
6. Create a FILESTREAM filegroup for the database. For example, to create a FILESTREAM filegroup named FileStreamFileGroup for the Litware2012 database, use the following query:

```
USE master
GO
```

```
ALTER DATABASE Litware2012 ADD
FILEGROUP FileStreamFileGroup CONTAINS FILESTREAM;
GO
```

7. Add FILESTREAM files to the FILESTREAM filegroup by specifying a folder location that does not currently exist. For example, to create and associate the C:\FSTRM directory with the FILESTREAM file named FileStrmFile in the FileStreamFileGroup FILESTREAM filegroup for the Litware2012 database, use the following query:

```
USE master
GO
ALTER DATABASE Litware2012 ADD FILE (
NAME = FileStrmFile,
FILENAME = 'C:\FSTRM')
TO FILEGROUP FileStreamFileGroup
```

MORE INFO **FILESTREAM**

You can learn more about FILESTREAM at *http://msdn.microsoft.com/en-us/library/gg471497(SQL.110).aspx*.

Configuring FileTables

FileTables are a special type of table that enables you to store files and documents within SQL Server 2012. These files and documents can be accessed from Windows applications as though they were stored normally in the file system. For example, you can add files and folders to the FileTable by dragging and dropping them in Windows Explorer. You can remove them from the FileTable by using the same method.

A FileTable provides the following functionality:

- A FileTable provides a hierarchy of files and directories.
- Each row in a FileTable represents a file or directory.
- Each row holds the following items:
  - A FILESTREAM column for stream data and a file_id (GUID) identifier.
  - Path_locator and parent_path_locator columns. These represent the file and directory hierarchy.
  - Ten file attributes. These include creation data and modification date.
  - Type column that supports full-text and semantic search.
- You can update FileTables by using normal Transact-SQL queries.

To enable FileTables, perform the following steps:

1. Enable FILESTREAM at the instance level. You can do this with the following query:

```
EXEC sp_configure filestream_access_level, 2
RECONFIGURE
```

2. Enable Non-Transactional Access at the database level. You can do this when creating a new database by using the CREATE DATABASE statement and the FILESTREAM NON_TRANSACTED_ACCESS option. For example:

```
CREATE DATABASE database_name
WITH FILESTREAM ( NON_TRANSACTED_ACCESS = FULL, DIRECTORY_NAME = N'dir_name')
```

You can do this for an existing database by using the ALTER DATABASE statement with the SET FILESTREAM option. For example:

```
ALTER DATABASE database_name
SET FILESTREAM (NON_TRANSACTED_ACCESS = FULL, DIRECTORY_NAME = N'directory_name')
```

3. Specify a Directory for FileTables at the database level if you haven't already done so when configuring Non-Transactional Access. You can modify the directory name by using the ALTER DATABASE statement with the SET FILESTREAM option. You can also configure the directory name on the Options page of the Database Properties dialog box, as shown in Figure 3-12.

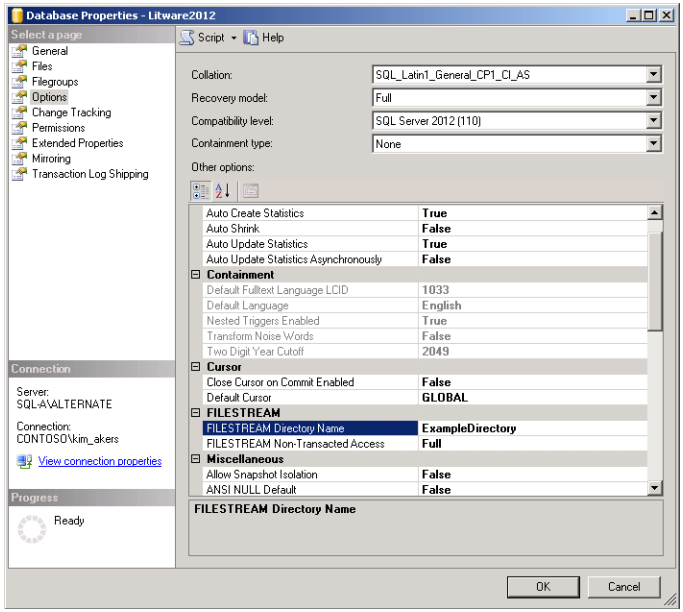

FIGURE 3-12 Configuring directory for FileTables

To create a FileTable by using SQL Server Management Studio, right-click the Tables node and choose New FileTable to open a new script window that contains a template Transact-SQL script that you can modify. You can also create a FileTable by using the CREATE TABLE statement with the AS FileTable option. For example, to create a new FileTable named DocStore, use the following query:

```
CREATE TABLE DocStore as FileTable;
GO
```

You can create FileTables subject to the following conditions:

- You cannot convert an existing table into a FileTable.
- You have specified a parent directory at the database level.
- A valid FILESTREAM filegroup exists. If you don't specify a filegroup, the default FILESTREAM filegroup will be used.
- You cannot create a table constraint when creating the table, but you can add one after the table is created.
- You cannot create a FileTable in the tempdb database.

FileTable tables have predefined and fixed schema, so it is not possible to add or change columns. It is possible to add custom indexes, triggers, and constraints to a FileTable. Dropping a FileTable also drops the directory and the subdirectories that it contained.

MORE INFO **FILETABLES**

You can learn more about FileTables at *http://msdn.microsoft.com/en-us/library/ff929144(SQL.110).aspx*.

PRACTICE Install Analysis Services and Reporting Services

In this practice, you deploy two instances of Analysis Services in different configurations and deploy an instance of Reporting Services.

EXERCISE 1 Install Analysis Services

In this exercise, you install two Analysis Services instances. The first Analysis Services instance will use multidimensional and data mining modes. The second Analysis Services instance will use tabular mode. To complete this exercise, perform the following steps:

1. Log on to server SQL-A with the Contoso\Kim_Akers user account.
2. Use the command line to install a new instance of Analysis Services on server SQL-A. The server should have the following properties:
   - Installation mode: Multidimensional and data mining mode
   - Instance name: ASMulti
   - Analysis Services service account: NetworkService
   - Analysis Services Server administrator: contoso\kim_akers
3. Use the command line to install an additional new instance of Analysis Services on server SQL-A. This instance of Analysis Services should have the following properties:
   - Installation mode: Tabular mode
   - Instance name: ASTabular
   - Analysis Services service account: NetworkService
   - Analysis Services Server administrator: contoso\kim_akers

EXERCISE 2 Install Reporting Services

In this exercise, you use Windows PowerShell to install the web server role that Reporting Services uses and then deploy a new Reporting Services instance from the command line. To complete this exercise, perform the following steps:

1. On SQL-A, open an elevated PowerShell prompt and run the following command:

   ```
   Import-module ServerManager
   ```

2. Run the following command:

   ```
   Add-WindowsFeature Web-Server -IncludeAllSubFeature
   ```

3. Install a new instance of Reporting Services from the command line. Use the following options so that you don't have to perform post-installation configuration by using Reporting Services Configuration Manager.
   - Install the SQL, Reporting Services, and Tools features.
   - Use **RPTSVR** as the name of the instance.
   - Use the NetworkService account for the Reporting Services, SQL Server, and SQL Server Agent service accounts.
   - Set the Reporting Services startup type to Manual.
   - Use the Default Native Mode Reporting Services installation mode.
   - Configure the BUILTIN\ADMINISTRATORS group for the SQL sysadmin accounts.
4. When the installation has finished, open Reporting Services Configuration Manager and verify that Reporting Services has deployed correctly and the service has started.

Lesson Summary

- An Analysis Services instance in multidimensional and data mining mode supports OLAP databases.
- An Analysis Services instance in tabular mode supports the new tabular modeling feature.
- If you perform a file-only Reporting Services deployment, you must run the Reporting Services Configuration Manager.
- FILESTREAM enables you to store BLOB objects in the file system.
- FileTables are special types of tables that enable you to store files and directories directly in the database. These files and directories can be accessed through the Windows file system.
- Analysis Services and Reporting Services can be enhanced with SharePoint integration.
- You configure Integration Services security with the DCOM Configuration Tool.

Lesson Review

Answer the following questions to test your knowledge of the information in this lesson. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the "Answers" section at the end of this chapter.

1. Which tool do you use to give a user access to Integration Services?
   - **A.** SQL Server Management Studio
   - **B.** SQL Server Configuration Manager
   - **C.** SQL Server Data Tools
   - **D.** DCOM Configuration Tool (Dcomcnfg.exe)
2. Which tool do you use to change the Reporting Services execution account?
   - **A.** SQL Server Management Studio
   - **B.** Reporting Services Configuration Manager
   - **C.** SQL Server Configuration Manager
   - **D.** SQL Server Installation Center
3. What is the maximum number of full-text indexes that you can configure for a partitioned table?
   - **A.** 1
   - **B.** 32
   - **C.** 1,024
   - **D.** 2,048
4. Which of the following steps must you take to enable FILESTREAM on a SQL Server 2012 instance that has both the Database Engine and Analysis Services features installed? (Each correct answer presents part of the solution. Choose two.)
   - **A.** Edit the properties of the SQL Server service in SQL Server Configuration Manager.
   - **B.** Edit the properties of the Analysis Services service in SQL Server Configuration Manager.
   - **C.** Run sp_configure filestream_access_level, 2.
   - **D.** Run sp_configure filestream_access_level, 0.

Lesson 2: Managing and Configuring Databases

Database administrators are often responsible for configuring and managing database infrastructure, such as the location of database files, and issues such as database encryption, which do not have a direct impact on users of the database. This lesson covers managing and configuring the properties of databases, including filegroups, database standardization, contained databases, data compression, transparent data encryption, partitioning, log file management, and database console commands.

After this lesson, you will be able to:

- Design and manage filegroups.
- Standardize and configure databases.
- Implement and configure contained databases.
- Configure data compression.
- Manage Transparent Data Encryption.
- Configure table and index partitioning.
- Manage the growth of log files.
- Use database console commands.

Estimated lesson time: 60 minutes

Designing and Managing Filegroups

Each database has a *primary filegroup*. This filegroup hosts the primary data file and any secondary files that you have not allocated to other filegroups. The system tables are hosted in the primary filegroup. You can create *filegroups* to host data files together for reasons including data allocation, administrative, and placement. Secondary data files use the .ndf extension. You can assign these secondary files to different filegroups. Secondary data files that are hosted on different volumes can be assigned to the same filegroup.

When you create a table or an index, you can configure it to use a specific filegroup. When a filegroup contains more than one file, the Database Engine will write data across the files proportionally, depending on how much free space is available in each file.

MORE INFO **FILEGROUPS**

You can learn more about filegroups at *http://msdn.microsoft.com/en-us/library/ms189563(SQL.110).aspx*.

Adding New Filegroups

To add a new filegroup to a database by using SQL Server Management Studio, perform the following steps:

1. In SQL Server Management Studio, right-click the database to which you want to add the filegroup and then choose Properties.
2. On the Filegroups page, click Add and then enter the name of the new filegroup, as shown in Figure 3-13.

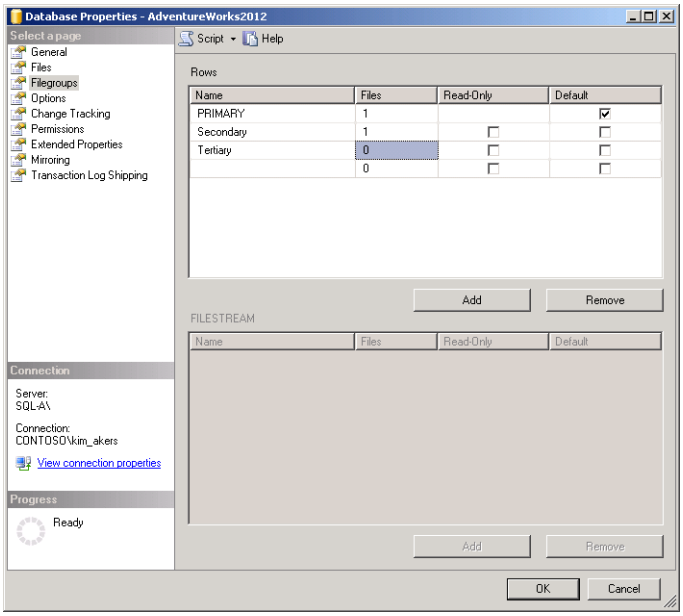

FIGURE 3-13 Adding a filegroup

3. You can add new files to a filegroup on the Files page in the Database Properties dialog box.

You can use the ALTER DATABASE statement with the ADD FILEGROUP option to add filegroups to a database. For example, to add a new filegroup named Tertiary to the AdventureWorks2012 database, use the following query:

```
USE [master]
GO
ALTER DATABASE [AdventureWorks2012] ADD FILEGROUP [Tertiary]
GO
```

Moving an Index from One Filegroup to Another

To move an index to a different filegroup or partition scheme in SQL Server Management Studio, perform the following steps:

1. In SQL Server Management Studio, right-click the index that you want to move to a new filegroup and then choose Properties.
2. On the Storage page, use the Filegroup drop-down list to select the filegroup to which you want to move the index, as shown in Figure 3-14.

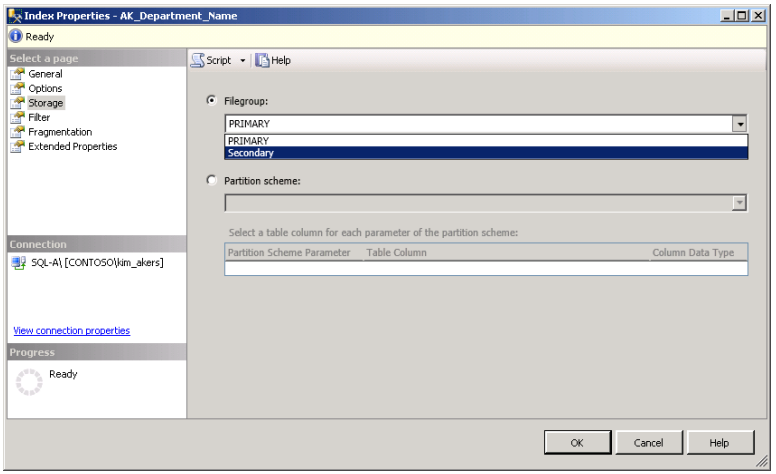

FIGURE 3-14 Moving an index

You can move indexes under the following conditions:

- You cannot move indexes created using a unique or primary key constraint through SQL Server Management Studio. You can move these indexes by using the CREATE INDEX statement with the (DROP_EXISTING=ON) option.
- If the table or index is partitioned, you must select the partition scheme in which to move the index.
- You can move clustered indexes by using online processing, allowing user access to data during the move.

To move an index by using Transact-SQL, use the CREATE INDEX statement with the DROP_EXISTING = ON option and specify the target filegroup.

> ***MORE INFO*** **MOVING AN EXISTING INDEX**
>
> **You can learn more about moving indexes to different filegroups at *http://msdn.microsoft.com/en-us/library/ms175905(SQL.110).aspx*.**

Configuring and Standardizing Databases

You can standardize the configuration of databases by configuring appropriate settings such as Auto Close, Auto Shrink, and database recovery model on the model system database. As you learned in Chapter 2, the model system database serves as a template for new databases that you create on an instance. You can configure the properties of the model database either by using SQL Server Management Studio, as shown in Figure 3-15, or by using the ALTER DATABASE statement.

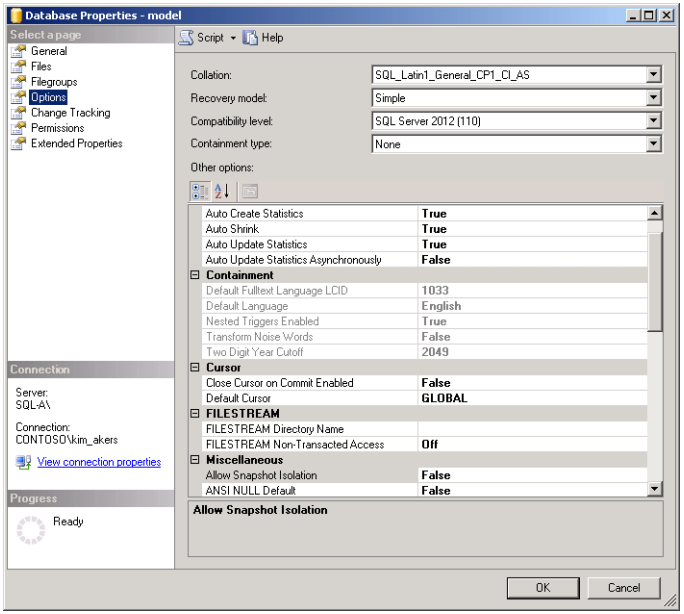

FIGURE 3-15 Model database properties

> ***MORE INFO*** **MODEL DATABASE**
>
> **You can learn more about configuring the model database at *http://msdn.microsoft.com/en-us/library/ms186388.aspx*.**

Understanding Contained Databases

Contained databases include all the settings and metadata required to define the database. Contained databases have no configuration dependencies on the Database Engine instance on which the database is deployed, so users connect to a contained database without authenticating at the Database Engine level. An advantage of contained databases is that you can easily move them to other instances or to SQL Server 2012 Azure. Having all database configuration settings within the database enables the database owners to manage all those settings for the database.

SQL Server 2012 supports contained databases and *Partially Contained Databases (Partial-CDBs)*, which provide a high degree of isolation from the Database Engine instance but are not fully contained. Partial-CDBs are a transitional step toward contained databases.

The SQL Server 2012 implementation of Partial-CDBs does not allow the following:

- Numbered procedures
- Schema-bound objects that depend on built-in functions with collation changes
- Binding change resulting from collation changes, including references to objects, columns, symbols, or types
- Replication, change data capture, and change tracking

You can use the sys.dm_db_uncontained_entities and sys.sql_modules (Transact-SQL) views to find information about uncontained objects or features. Through these views, you can determine the containment status of applications and work out which objects or features you must replace or modify when transitioning to a fully contained database. You should also monitor the database_uncontained_usage event to determine whether uncontained features are used in a database.

You can enable contained databases by using SQL Server Management Studio on the Advanced page of an instance's properties by setting the Enable Contained Databases option to True, as shown in Figure 3-16.

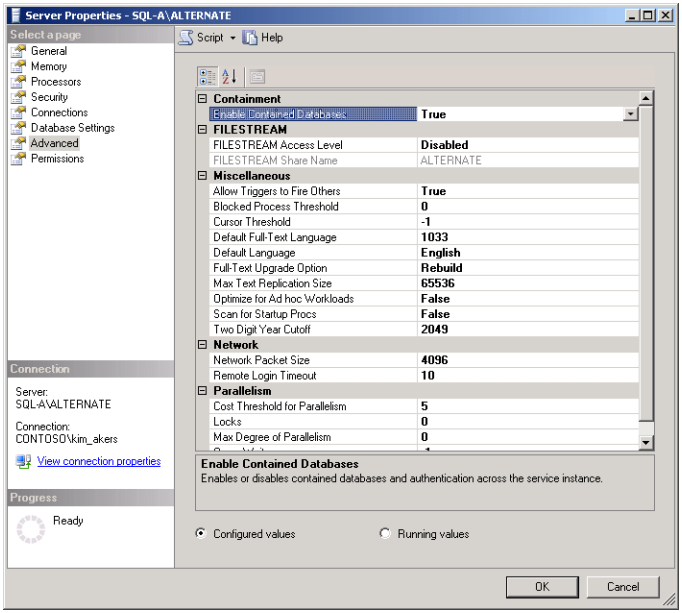

FIGURE 3-16 Enabling contained databases

To enable contained databases on an instance of SQL Server 2012 by using Transact-SQL, issue the following query:

```
sp_configure 'contained database authentication', 1;
GO
RECONFIGURE ;
GO
```

To convert a database to a Partial-CDB or contained database by using SQL Server Management Studio, edit the properties of a database and, on the Options page, change the Containment Type option to Partial or Full, as shown in Figure 3-17.

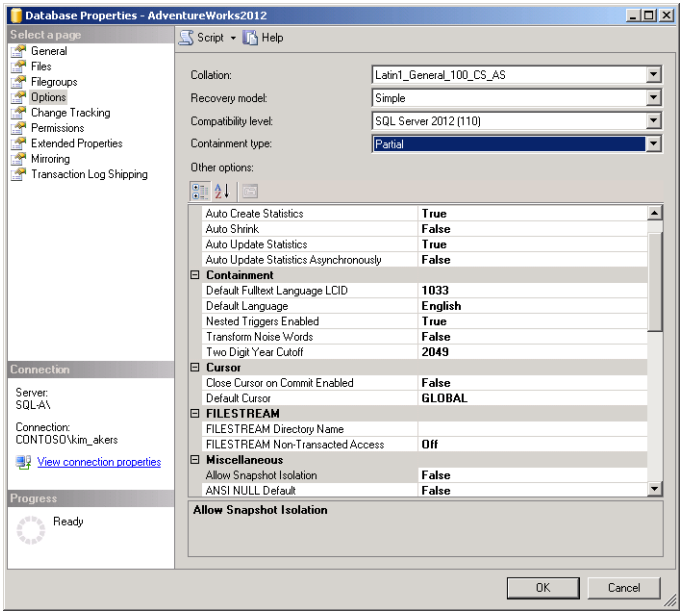

FIGURE 3-17 Database containment type

You can use the ALTER DATABASE statement with the SET CONTAINMENT option to configure containment for a database after you have enabled containment at the database instance level. For example, to set the containment of the AdventureWorks2012 database to Partial, use the following query:

```
USE [master]
GO
ALTER DATABASE [AdventureWorks2012] SET CONTAINMENT = PARTIAL
GO
```

MORE INFO **PARTIALLY CONTAINED DATABASES**

You can learn more about Partially Contained Databases at *http://msdn.microsoft.com/en-us/library/ff929071(SQL.110).aspx*.

Using Data Compression

Row and page compression for tables and indexes enables you to save storage space by reducing the size of the database. Data compression has the drawback of increasing CPU usage because the data must be compressed and decompressed when being accessed. You cannot use data compression with system tables, and only the Enterprise and Developer editions of SQL Server 2012 support data compression.

You can configure data compression on the following:

- Clustered tables
- Heap tables (a heap is a table without a clustered index)
- Non-clustered indexes
- Indexed views
- Individual partitions of a partitioned table or index

There are three forms of data compression you can use with SQL Server 2012: row-level compression, unicode compression, and page-level compression.

> ***MORE INFO*** **HEAPS**
>
> **You can learn more about heaps at *http://msdn.microsoft.com/en-us/library/hh213609(v=SQL.110).aspx*.**

Row-Level Compression

Row-level compression works by using more efficient storage formats for fixed-length data. Row-level compression uses the following strategies to save space:

- Storing fixed-length numeric data types and CHAR data types as though they were variable-length data types
- Not storing NULL or 0 values
- Reducing metadata required to store data

Although it does reduce the amount of space that data uses, row-level compression does not provide the storage improvements of page-level compression. The advantage of row-level compression is that it requires less CPU usage than page-level compression. You use the following syntax to compress a table by using row-level compression:

```
ALTER TABLE tableName REBUILD WITH (DATA_COMPRESSION=ROW)
```

For example, to rebuild all partitions of the Sales.Customer table of the AdventureWorks2012 database by using row compression, use the following query:

```
USE [AdventureWorks2012]
ALTER TABLE [Sales].[Customer] REBUILD PARTITION = ALL
WITH (DATA_COMPRESSION = ROW)
```

You use the following syntax to configure an index with row-level compression:

```
ALTER INDEX indexName ON tableName REBUILD PARTITION ALL WITH (DATA_COMPRESSION=ROW)
```

> ***MORE INFO*** **ROW-LEVEL COMPRESSION**
>
> **You can learn more about row-level compression at *http://msdn.microsoft.com/en-us/library/cc280576(v=sql.110).aspx*.**

Unicode Compression

Unicode compression enables the database engine to compress unicode values stored in page or row compressed objects. You can use unicode compression with the fixed-length nchar(n) and nvarchar(n) data types. Unicode compression is automatically used where appropriate when you enable row and page compression.

> ***MORE INFO*** **UNICODE COMPRESSION**
>
> **You can learn more about unicode compression at *http://msdn.microsoft.com/en-us/library/ee240835(SQL.110).aspx*.**

Page-Level Compression

Page-level compression compresses data by storing repeating values and common prefixes only once and then making references to those values from other locations within the table. When page compression is applied to a table, row compression techniques are also applied. Page-level compression uses the following strategies:

- Row-level compression is applied to maximize the number of rows stored on a page.
- Column prefix compression is applied by replacing repeating data patterns with references. This data is stored in the page header.
- Dictionary compression scans for repeating values and then stores this information in the page header.

The benefits of page compression depend on the type of data compressed. Data that involves many repeating values will be more compressed than data populated by more unique values. You use the following general syntax to apply page-level compression:

```
ALTER TABLE name REBUILD WITH (DATA_COMPRESSION=PAGE)
```

For example, to rebuild all partitions of the Sales.Customer table of the AdventureWorks2012 database by using page compression, use the following query:

```
USE [AdventureWorks2012]
ALTER TABLE [Sales].[Customer] REBUILD PARTITION = ALL
WITH
(DATA_COMPRESSION = PAGE)
```

You use the following syntax to configure an index with page-level compression:

```
ALTER INDEX indexName ON tableName REBUILD PARTITION ALL WITH (DATA_COMPRESSION=PAGE)
```

> ***MORE INFO*** **PAGE-LEVEL COMPRESSION**
>
> **You can learn more about page-level compression at *http://msdn.microsoft.com/en-us/library/cc280464(v=sql.110).aspx*.**

If tables or indexes are partitioned, you can configure compression on a per-partition basis. If you split a partition by using the ALTER PARTITION statement, the new partitions inherit the data compression attribute of the original partition. If you merge two partitions, the resulting partition has the compression attribute of the destination partition.

Although compression does allow more rows to be stored on a page, it doesn't alter the maximum row size of a table or index. You can't enable a table for compression if the maximum row size and the compression overhead exceed 8,060 bytes.

The default compression setting for indexes is NONE, and you must specify the compression property for indexes when you create them. Non-clustered indexes do not inherit the compression property of the table, but clustered indexes created on a heap inherit the compression state of the heap.

Data compression applies only at the source, so when you export data from a compressed source, SQL Server will output the data in uncompressed row format. Importing uncompressed data into a target table enabled for compression will compress the data.

> ***MORE INFO*** **DATA COMPRESSION**
>
> **You can learn more about data compression at *http://msdn.microsoft.com/en-us/library/cc280449(v=sql.110).aspx*.**

You can configure compression by using the preceding Transact-SQL statements or from SQL Server Management Studio by using the Data Compression Wizard on either tables or indexes. You can use the Data Compression Wizard to add and remove compression. To use the Data Compression Wizard to change the compression settings for both tables and indexes, perform the following steps:

1. In SQL Server Management Studio, right-click the table or index you want to compress, choose Storage, and then select Manage Compression.
2. On the Welcome To The Data Compression Wizard page, click Next.
3. On the Select Compression Type page, shown in Figure 3-18, you can choose to use the same compression type for all partitions or choose among Row, Page, and None on a per-partition basis. Click Calculate to determine the difference between current space usage and compressed usage.

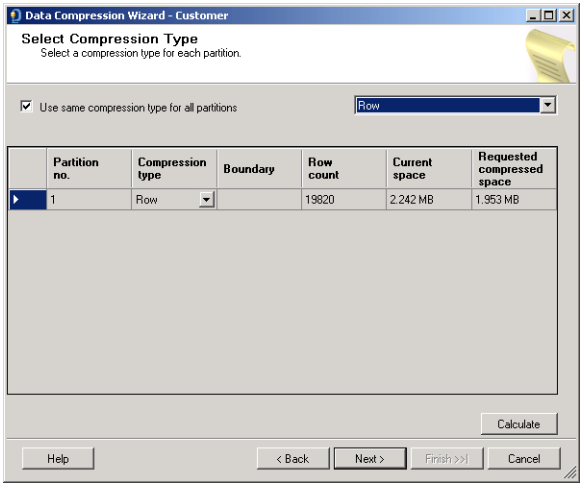

FIGURE 3-18 Data Compression Wizard

4. On the Select An Output Option page, choose whether to create a script, to perform the operation immediately, or to perform the option according to a schedule. Click Next and then click Finish to complete the wizard.

> ***MORE INFO*** **DATA COMPRESSION WIZARD**
>
> **You can learn more about the Data Compression Wizard at *http://msdn.microsoft.com/en-us/library/cc280496(v=SQL.110).aspx*.**

Estimating Compression

The best way to determine the benefits of compression on an object is to use the sp_estimate_data_compression_savings stored procedure. The benefits of compression depend on factors such as the uniqueness of data. The sp_estimate_data_compression_savings stored procedure is available in the Enterprise edition of SQL Server 2012 only.

The syntax of the stored procedure is as follows:

```
sp_estimate_data_compression_savings[ @schema_name = ] 'schema_name', [ @object_name = ]
'object_name', [@index_id = ] index_id,[@partition_number = ] partition_number,
[@data_compression = ] 'data_compression'
```

For example, to configure an estimate of the compression benefits of using Row compression on the HumanResources.Employee table in the AdventureWorks2012 database, execute the following Transact-SQL statement:

```
USE AdventureWorks2012;
GO
EXEC sp_estimate_data_compression_savings 'HumanResources', 'Employee', NULL, NULL,
'ROW';
GO
```

To configure an estimate of the compression benefits of using Page compression on the same table, execute the following Transact-SQL statement:

```
USE AdventureWorks2012;
GO
EXEC sp_estimate_data_compression_savings 'HumanResources', 'Employee', NULL, NULL,
'PAGE';
GO
```

MORE INFO **SP_ESTIMATE_DATA_COMPRESSION_SAVINGS**

You can learn more about how to estimate compression savings at *http://msdn.microsoft.com/en-us/library/cc280574(v=sql.110).aspx*.

Encrypting Databases with Transparent Data Encryption

Transparent Data Encryption (TDE) enables you to encrypt an entire database. TDE protects the database against unauthorized third parties gaining access to the hard disks or backups on which the database is stored. TDE encrypts the database by using a Database Encryption Key (DEK) that is stored in the database boot record. The DEK is in turn protected by the database master key, which is in turn protected by the service master key. You can use BitLocker Drive Encryption, a full-volume encryption method supported by Windows Server 2008 and Windows Server 2008 R2, although this will not ensure that database backups are encrypted.

NOTE **TDE AND TEMPDB**

If any database on the instance uses TDE, the tempdb system database will also be encrypted.

To use TDE to encrypt a database, you must perform the following steps:

1. Create the master encryption key.
2. Create the certificate protected by the master key.
3. Create a DEK and protect it by using the certificate.
4. Encrypt the database.

The first step in deploying TDE involves creating a master encryption key. You do this by using the CREATE MASTER KEY ENCRYPTION BY PASSWORD statement. For example, you can accomplish that by using the following query:

```
USE master;
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<MasterKeyPasswordHere>';
GO
```

After you have created the master encryption key, the next step involves creating the certificate that will be used to encrypt the database. You can accomplish this by using the CREATE CERTIFICATE statement. For example, to create a certificate named ServerCertificate that uses the subject name Server Certificate, use the following query:

```
CREATE CERTIFICATE ServerCertificate WITH SUBJECT = 'Server Certificate';
GO
```

When the master key and certificate are in place, you can create the DEK for the specific database. You do this by using the CREATE DATABASE ENCRYPTION KEY statement. For example, the following query creates a DEK for the AdventureWorks2012 database:

```
USE AdventureWorks2012;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_128
ENCRYPTION BY SERVER CERTIFICATE ServerCertificate;
GO
```

After all the appropriate keys and certificates are in place, you can encrypt the database by using the ALTER DATABASE statement. For example, to encrypt the AdventureWorks2012 database, use the following query:

```
ALTER DATABASE AdventureWorks2012
SET ENCRYPTION ON;
GO
```

When using TDE, you should create a backup of the server certificate in the master database. If you lose the database server without backing this up, you cannot access data in a database protected by TDE. You can use the BACKUP CERTIFICATE statement to create a backup of the certificate and private key, both of which are required for certificate recovery. The private key password does not have to be the same as the database master key password. For example, the following code, when run from the master system database, creates a backup of the ServerCertificate certificate to a file called ServerCertExport and a PrivateKeyFile private key:

```
BACKUP CERTIFICATE ServerCertificate
TO FILE = 'ServerCertExport'
WITH PRIVATE KEY
(
    FILE = 'PrivateKeyFile',
    ENCRYPTION BY PASSWORD = '<PrivateKeyPasswordHere>'
);
GO
```

SQL Server will write these backup files to the \MSSQL\DATA directory of the instance.

MORE INFO **TRANSPARENT DATA ENCRYPTION**

You can learn more about Transparent Data Encryption at *http://msdn.microsoft.com/en-us/library/bb934049(SQL.110).aspx*.

Quick Check

- **After the appropriate keys and certificates are created, which Transact-SQL statement do you use to encrypt a database?**

Quick Check Answer

- **Use ALTER DATABASE with the SET ENCRYPTION ON option.**

Partitioning Indexes and Tables

Partitioning divides index and table data across more than one filegroup. Data is partitioned so that groups of rows are mapped to individual partitions. All partitions of a table or index must reside in the same database. You can use partitioned indexes and tables only in the Enterprise and Developer editions of SQL Server 2012. The x64 versions of SQL Server 2012 support up to 15,000 partitions. It is possible to create more than 1,000 partitions on the x86 versions of SQL Server 2012, but this is not supported by Microsoft.

Table and index partitioning involves the following concepts:

- **Partition function** Defines how the rows of an index or table map to specific partitions based on the values of partitioning columns
- **Partition scheme** Maps the partitions of a partition function to a collection of filegroups
- **Partitioning column** The column of an index or table that a partition function uses to partition the index or table
- **Aligned index** An index that uses the same partition scheme as the table to which it belongs
- **Nonaligned index** An index that is partitioned independently from the table to which it belongs
- **Partition elimination** The process through which the query optimizer will access only the appropriate partitions to satisfy a query's filter criteria

When creating a partitioned table or index, follow these general steps:

1. Create the filegroup or filegroups that will host the partitions.
2. Create a partition function that assigns rows of a table or index to partitions based on the values of a specific column.
3. Create a partition scheme that maps partitions to filegroups.
4. Create or modify an index or table and specify the partition scheme.

Partitioned tables or indexes require the following permissions:

- To create a partitioned table, a user needs the CREATE TABLE permission in the database, the ALTER permission on the schema in which the table is being created, and one of the following permissions:
  - ALTER ANY DATASPACE permission
  - CONTROL or ALTER permission on the database
  - CONTROL SERVER or ALTER ANY DATABASE permission on the instance
- To create a partitioned index, a user needs the ALTER permission on the table or view that hosts the index and one of the following permissions:
  - ALTER ANY DATASPACE permission
  - CONTROL or ALTER permission on the database
  - CONTROL SERVER or ALTER ANY DATABASE permission on the instance

To create a partitioned table, perform the following steps:

1. In SQL Server Management Studio, right-click the table you want to partition, choose Storage, and then select Create Partition.
2. On the first page of the Create Partition Wizard, click Next.
3. On the Select A Partitioning Column page, shown in Figure 3-19, choose the column you will use to partition the table. You can also enable the following options:

- **Collocate This Table To The Selected Partitioned Table** This option can improve query efficiency.
- **Storage-Align Non-Unique Indexes And Unique Indexes With Indexed Partitioning Column** This option enables you to move partitions in and out of partitioned tables more effectively.

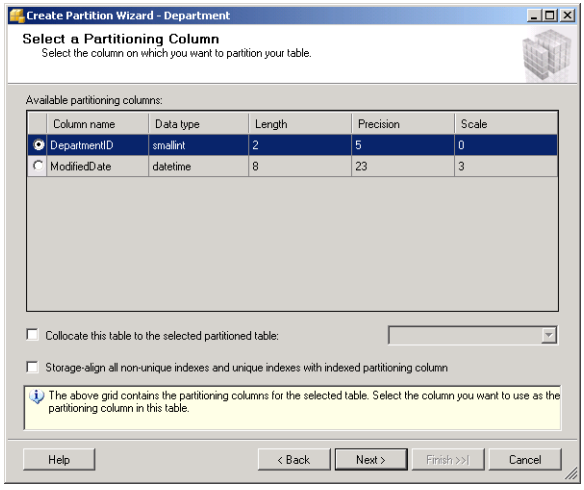

FIGURE 3-19 Select a partitioning column

4. On the Select A Partition Function page, either choose an existing partition function or specify the name of a new partition function.
5. On the Select A Partition Scheme page, either choose an existing partition scheme or enter the name of a new partition scheme.
6. On the Map Partitions page, choose Left Boundary or Right Boundary to determine whether to include the highest or lowest bounding value within each filegroup. In the Boundary column, specify the boundary value, as shown in Figure 3-20. You can also choose Set Boundaries if you want to use date values with a partitioning column. This is useful for separating table data into filegroups based on date, but it is available only if the partitioning column is of type date, datetime, smalldatetime, datetime2, or datetimeoffset.

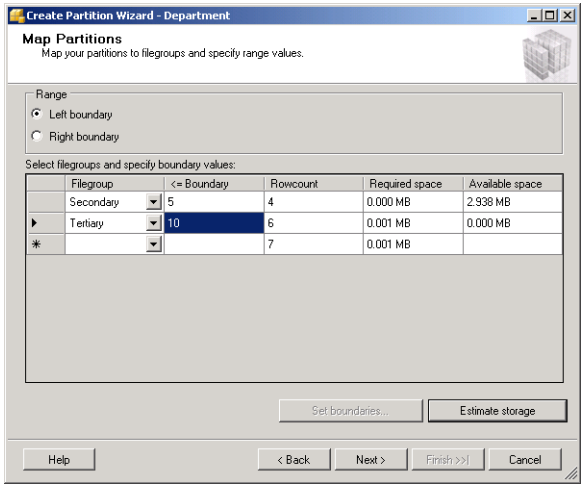

FIGURE 3-20 Map partitions

7. On the Select An Output Option page, choose between Create a Script or Run Immediately or to run on a schedule.
8. On the Review Summary page, click Finish.

You can create a partition function by using the CREATE PARTITION FUNCTION statement. For example, to create a function named PartFunction that will divide a table into two partitions by using the number 50, use the following statement:

```
CREATE PARTITION FUNCTION PartFunction (int)
as RANGE LEFT FOR VALUES (50);
```

You can create a partition scheme with the CREATE PARTITION SCHEME statement. For example, to create a partition scheme named PartScheme that applies the partition function PartFunction to the filegroups FgOne and FgTwo, use the following query:

```
CREATE PARTITION SCHEME PartScheme
       AS PARTITION PartFunction
```

```
        TO (FgOne, FgTwo);
GO
```

You can create a partitioned table by referencing the partition scheme. For example, to create a table called Exemplar that uses the PartScheme partition scheme to partition col1, use the following query:

```
CREATE TABLE Exemplar (col1 int, col2 char(20))
        ON PartScheme (col1);
GO
```

You can use the ALTER PARTITION FUNCTION statement to modify an existing partition function by either splitting one partition into two or merging two partitions into one. You can use the ALTER PARTITION SCHEME statement to modify an existing partition scheme.

MORE INFO **PARTITIONED INDEXES AND TABLES**

You can learn more about partitioned tables and indexes at *http://msdn.microsoft.com/en-us/library/ms190787(SQL.110).aspx*.

Managing Log Files

Transaction log files use the .ldf extension. Although a database might have multiple log files, SQL Server treats these multiple log files as a single contiguous-file virtual log file. If log records were never deleted, the logs would grow to consume the volume on which they were hosted. Log truncation is the process by which the Database Engine frees space in the logical log for reuse by the transaction log. Log truncation occurs automatically in the following situations:

- When a database uses the simple recovery model, the Database Engine truncates the transaction log after a checkpoint. Automatic checkpoints are triggered each time the number of log records reaches the number that the Database Engine determines it can process during the *recovery interval* server configuration option. The Database Engine triggers an automatic checkpoint under the simple recovery model when the virtual log becomes 70 percent full. You can trigger checkpoints manually by using the CHECKPOINT statement.
- When a database uses the full recovery model or bulk-logged recovery model, the Database Engine truncates the transaction log after a log backup as long as a checkpoint has occurred since the previous backup. You will learn more about backing up transaction logs in Chapter 11, "Backup and Restore."

You can use the DBCC SQLPERF (LOGSPACE) command to monitor the amount of log space used. Figure 3-21 shows the output of this command.

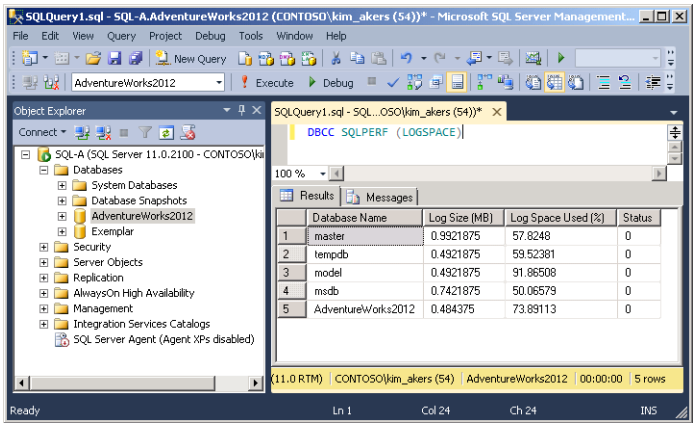

FIGURE 3-21 DBCC SQLPERF (LOGSPACE)

You can add log files to the database by using the ALTER DATABASE statement with the ADD LOG FILE option. You can modify the size of a transaction log file by using the ALTER DATABASE statement with the MODIFY FILE option and specifying a SIZE and MAXSIZE figure. By default, SQL Server 2012 transaction log files are configured to autogrow by 10 percent to a maximum of 2,097,152 MB.

MORE INFO **TRANSACTION LOG FILES**

You can learn more about the transaction log at *http://msdn.microsoft.com/en-us/library/ms190925(SQL.110).aspx*.

Using Database Console Commands

Database console commands (DBCC) enable you to perform SQL Server 2012 administration tasks by using queries rather than by using the SQL Server Management Studio graphical user interface (GUI). DBCC commands are grouped into the following categories:

- **Maintenance** Commands that perform maintenance tasks on databases, indexes, or filegroups
- **Informational** Commands that display SQL Server information
- **Validation** Commands that enable you to validate operations on databases, tables, indexes, catalogs, or filegroups
- **Miscellaneous** Commands that enable you to perform tasks such as enabling trace flags or modifying which dynamic-link libraries (DLLs) are loaded into memory

Maintenance Statements

DBCC maintenance statements, their functions, and the permissions required to run them are as follows:

- **DBCC CLEANTABLE** Reclaims space from dropped variable-length columns in indexed views or tables. The user must own the table or indexed view or be a member of the sysadmin fixed server role or the db_owner or db_ddladmin fixed database roles.
- **DBCC DBREINDEX** Rebuilds one or more indexes for a table. The user must be a member of the sysadmin fixed server role or the db_owner or db_ddladmin fixed database roles. This statement is deprecated in this version of SQL Server, and you should use ALTER INDEX to perform this task.
- **DBCC DROPCLEANBUFFERS** Removes all clean buffers from the buffer pool. The user must be a member of the sysadmin fixed server role.
- **DBCC FREEPROCCACHE** Removes all elements from the plan cache, specific plan from the plan cache, or all cache entries related to a specific resource pool. The user must have ALTER SERVER STATE permission in the Database Engine.
- **DBCC INDEXDEFRAG** Defragments indexes. This feature will be removed in future versions of SQL Server, and you should use ALTER INDEX instead. The user must be a member of the sysadmin fixed server role or the db_owner or db_ddladmin fixed database roles.
- **DBCC SHRINKDATABASE** Shrinks the size of all data and log files of the specified database. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role.
- **DBCC SHRINKFILE** Shrinks a specified data or log file. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role.
- **DBCC UPDATEUSAGE** Updates page and row count data for catalog views to remove inaccuracies. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role. This statement is provided for backward compatibility.

Informational Statements

DBCC informational statements, their functions, and the permissions required to run them are as follows:

- **DBCC INPUTBUFFER** Shows the last statement forward from a client to the Database Engine. The user must be a member of the sysadmin fixed server role or have the VIEW SERVER STATE permission.
- **DBCC OPENTRAN** Shows information about the oldest running transaction, oldest running distributed transaction, and oldest running non-distributed transaction. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role.

- **DBCC OUTPUTBUFFER** Displays the current output buffer in hexadecimal and ASCII format for a specific session_id. The user must be a member of the sysadmin fixed server role.
- **DBCC PROCCACHE** Provides information about the procedure cache. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role.
- **DBCC SHOW_STATISTICS** Provides current query optimization statistics for a table or indexed view. The user must be a member of the sysadmin fixed server role or the db_owner or db_ddladmin fixed database roles.
- **DBCC SHOWCONTIG** Provides fragmentation information for tables, views, or indexes. The user must be a member of the sysadmin fixed server role or the db_owner or db_ddladmin fixed database roles. This statement is deprecated, and you should migrate to using appropriate dynamic management views for this information.
- **DBCC SQLPERF** Displays transaction log space usage statistics for all databases hosted by an instance. Access requires the VIEW SERVER STATE permission on the server.
- **DBCC TRACESTATUS** Provides information about trace flags. The user must be a member of the public role.
- **DBCC USEROPTIONS** Provides information about currently set options on the connection. The user must be a member of the public role.

Validation Statements

DBCC validation statements, their functions, and the permissions required to run them are as follows:

- **DBCC CHECKALLOC** Performs a consistency check of disk space allocation structures. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role.
- **DBCC CHECKCATALOG** Checks catalog consistency of online databases. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role.
- **DBCC CHECKCONSTRAINTS** Verifies the integrity of a specific constraint or all constraints on a table within the current database. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role.
- **DBCC CHECKDB** Checks the physical and logical integrity of all objects in a specific database; runs DBCC CHECKALLOC, DBCC CHECKTABLE, and DBCC CHECKCATALOG; validates the contents of every indexed view, link-level consistency between table metadata, file system directories, and files when storing varbinary(max) data on the file system using FILESTREAM; and checks Service Broker data. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role.

MORE INFO **DBCC CHECKDB**

You can learn more about DBCC CHECKDB at *http://msdn.microsoft.com/en-us/library/ms176064(SQL.110).aspx*.

- **DBCC CHECKFILEGROUP** Verifies the allocation and structural integrity of indexed views and tables in a specific filegroup. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role.
- **DBCC CHECKIDENT** Verifies and, if necessary, changes the identity value for a specific table. The user must be a member of the sysadmin fixed server role or the db_owner or db_ddladmin fixed database roles.
- **DBCC CHECKTABLE** Verifies the integrity of all pages and structures that make up a table or indexed view. The user must be a member of the sysadmin fixed server role or the db_owner or db_ddladmin fixed database roles.

EXAM TIP

Be familiar with the functionality of DBCC CHECKDB prior to taking the 70-462 exam.

Miscellaneous Statements

Miscellaneous DBCC statements, their functions, and the permissions required to run them are as follows:

- **DBCC dllname (FREE)** Unloads a specific extended stored procedure DLL from memory. The user must be a member of the sysadmin fixed server role or the db_owner fixed database role.
- **DBCC FREESESSIONCACHE** Flushes the distributed query connection cache. The user must be a member of the sysadmin fixed server role.
- **DBCC FREESYSTEMCACHE** Flushes all unused cache entries from all caches. Access requires the ALTER SERVER STATE permission.
- **DBCC HELP** Provides information on a specific DBCC command. The user must be a member of the sysadmin fixed server role.
- **DBCC TRACEOFF** Disables specific trace flags. The user must be a member of the sysadmin fixed server role.
- **DBCC TRACEON** Enables specific trace flags. The user must be a member of the sysadmin fixed server role.

MORE INFO **DBCC**

You can learn more about DBCC at *http://msdn.microsoft.com/en-us/library/ms188796(SQL.110).aspx*.

PRACTICE Table Partitioning, Compression, Encryption, and Log Files

In this practice, you partition tables, configure encryption and compression, and manage SQL Server log files.

EXERCISE 1 Partition Tables

In this exercise, add filegroups and files to the WingTipToys2012 database. You create a partition function and a partition scheme and then create a table that uses that partition scheme. To complete this exercise, perform the following steps:

1. Log on to server SQL-A with the Kim_Akers user account.
2. Using SQL Server Management Studio, create a database named **WingTipToys2012** on instance SQL-A.
3. Add filegroups **FgOne**, **FgTwo**, and **FgThree** to database WingTipToys2012.
4. Add a database file named **file1** to FgOne, a database file named **file2** to FgTwo, and a database file named **file3** to FgThree.
5. Use Transact-SQL on the WingTipToys2012 database to create a partition function named **WTPartFunction** for the integer data type for RANGE LEFT values of 30 and 60.
6. Use Transact-SQL to create a partition scheme named **WTPartScheme** for WTPartFunction for FgOne, FgTwo, and FgThree.
7. Create a table named **Toys,** in which column 1 uses the integer data type and column 2 uses the char(30)data type, by using the WTPartScheme on column 1.

EXERCISE 2 Configure Encryption and Compression

In this exercise, connect to the SQL-A\ALTERNATE instance, create a new database, encrypt that new database, and then create new tables and configure them with compression. To complete this exercise, perform the following steps:

1. Use SQL Server Management Studio to connect to instance SQL-A\Alternate.
2. Create a new database named **WingTipToys2012** by using the default settings.
3. Use appropriate Transact-SQL queries to encrypt the WingTipToys2012 database by using Transparent Database Encryption.
4. Create a table in the WingTipToys2012 database named **Aeroplanes**. The table should have a single column named **Model** and should use the varchar(max) data type.
5. Configure the Aeroplanes table to use row-level compression.
6. Create a table in the WingTipToys2012 database named **Helicopters**. The table should have a single column named **Model** and should use the varchar(max) data type.
7. Configure the Helicopters table to use page-level compression.

EXERCISE 3 Manage Transaction Log Files

In this exercise, you manage transaction log files. To complete this exercise, perform the following steps:

1. On the default instance on SQL-A, add a file to the transaction log in the WingTipToys2012 database.
2. Trigger a transaction log file checkpoint in the WingTipToys2012 database.
3. Use the appropriate Transact-SQL code to determine how much free space is available in transaction logs.

Lesson Summary

- Filegroups are collections of database files that enable you to implement partitioning of tables and indexes.
- Configure the model system database as a template when standardizing databases on an instance.
- Contained databases are databases that have no dependencies on the Database Engine. This makes it easy to move databases between instances and to cloud-based deployments such as SQL Azure.
- Row-level compression modifies data types to reduce the amount of storage space used. Page-level compression uses dictionary compression techniques and provides greater space savings, but at the cost of CPU usage.
- Transparent Data Encryption (TDE) enables you to encrypt an entire database. The database will remain encrypted even when backed up.
- Transaction log truncation depends on the configured recovery model. You can force a checkpoint by using the CHECKPOINT statement.

Lesson Review

Answer the following questions to test your knowledge of the information in this lesson. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the "Answers" section at the end of this chapter.

1. Which statement would you use to add a filegroup to an existing database?
   - **A.** ALTER DATABASE
   - **B.** CREATE DATABASE
   - **C.** ALTER TABLE
   - **D.** CREATE TABLE
2. The STUDENTS table contains name, address, and contact information for students at a local college. Columns include Student_Name, DOB, Telephone, Email, Street_Address, Town, State, and Zip Code. IDX1 is an index on the Student_Name column. Given this

information, which of the following statements will provide the greatest reduction in the amount of space required to store data for the STUDENTS table?

A. ALTER TABLE STUDENTS REBUILD WITH (DATA_COMPRESSION=ROW)

B. ALTER TABLE STUDENTS REBUILD WITH (DATA_COMPRESSION=PAGE)

C. ALTER INDEX IDX1 ON STUDENTS REBUILD PARTITION ALL WITH (DATA_COMPRESSION=ROW)

D. ALTER INDEX IDX1 ON STUDENTS REBUILD PARTITION ALL WITH (DATA_COMPRESSION=PAGE)

3. Which of the following must you do before enabling Transparent Data Encryption for a database? (Each correct answer presents part of the complete solution. Choose three.)

A. Create a master encryption key.

B. Create a certificate.

C. Create a database encryption key.

D. Enable page-level compression.

4. Which command would you run if you wanted to check the physical and logical integrity of all objects within a specific database?

A. DBCC CHECKFILEGROUP

B. DBCC CHECKDB

C. DBCC SQLPERF

D. DBCC SHRINKDATABASE

Case Scenarios

In the following case scenarios, apply what you have learned about configuring additional SQL Server components and managing and configuring databases. You can find answers to these questions in the "Answers" section at the end of this chapter.

Case Scenario 1: Configuring FILESTREAM and FileTable

You have recently deployed SQL Server 2012 on a server named SYDNEY-DB. You want to use this server to store a large number of image files, most of which are between 10 MB and 20 MB in size. In view of this goal, you want to configure the default instance on SYDNEY-DB to support FILESTREAM. You also want to configure FileTables to simplify the process of adding image files to the database. With this information in mind, answer the following:

1. What general steps must you take to enable FILESTREAM on the default instance of server SYDNEY-DB?
2. After FILESTREAM is enabled on the default instance of server SYDNEY-DB, what general steps must you take to create a FileTable?

Case Scenario 2: Deploying Transparent Data Encryption

You want to deploy Transparent Data Encryption (TDE) to protect the ContosoCars2012 database hosted on one of your organization's SQL Server 2012 Database Engine instances. With that in mind, answer the following questions:

1. Which query would you use to create a master encryption key with the password **P@ssw0rd**?
2. Which query would you use to create the certificate that encrypts the database if the certificate name is ServerCertA and the subject name is Server Certificate A?
3. Which query would you use to create a Database Encryption Key (DEK) for the ContosoCars2012 database if you were using the AES_128 encryption algorithm and ServerCertA?
4. Which query would you use to encrypt the ContosoCars2012 database?
5. Which query would you use to back up the server certificate to a file named CertExport and the private key to a file named PrivateKey with the password **P@ssw0rd**?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following tasks.

FILESTREAM and FileTable

Prior to completing each task in the following practices, list the steps you would take to accomplish the task. After completing the task, assess how accurately you predicted the necessary steps.

- **Practice 1** Enable FILESTREAM on the ALTERNATE instance hosted on server SQL-A.
- **Practice 2** Create a new custom database on the ALTERNATE instance on server SQL-A. Create and populate a FileTable in this database.

Transparent Data Encryption and Table Partitioning

Prior to completing each task in the following practices, list the steps you would take to accomplish the task. After completing the task, assess how accurately you predicted the necessary steps.

- **Practice 1** Encrypt the new database on the ALTERNATE instance of SQL-A by using TDE.
- **Practice 2** Add a second filegroup and files to the new custom database. Create a partitioned table that uses the primary and second filegroup.

Answers

This section contains the answers to the lesson review questions and solutions to the case scenarios in this chapter.

Lesson 1

1. **Correct Answer: D**
   - **A. Incorrect.** You cannot use SQL Server Management Studio to give a user access to Integration Services.
   - **B. Incorrect.** You cannot use SQL Server Configuration Manager to give a user access to Integration Services.
   - **C. Incorrect.** SQL Server Data Tools enables you to create Analysis Services, Integration Services, and Report Server projects, but it cannot be used to give a user access to Integration Services.
   - **D. Correct.** Dcomcnfg.exe is the tool you can use to give non-administrative users access to Integration Services.
2. **Correct Answer: B**
   - **A. Incorrect.** You cannot use SQL Server Management Studio to change the Reporting Services execution account.
   - **B. Correct.** You use Reporting Services Configuration Manager to change the Reporting Services execution account.
   - **C. Incorrect.** You cannot use SQL Server Configuration Manager to change the Reporting Services execution account.
   - **D. Incorrect.** You cannot use SQL Server Installation Center to change the Reporting Services execution account.
3. **Correct Answer: A**
   - **A. Correct.** The maximum number of full-text indexes for a table is 1.
   - **B. Incorrect.** The maximum number of full-text indexes for a table is 1. You cannot have 32 full-text indexes for a partitioned table.
   - **C. Incorrect.** The maximum number of full-text indexes for a table is 1. You cannot have 1,024 full-text indexes for a partitioned table.
   - **D. Incorrect.** The maximum number of full-text indexes for a table is 1. You cannot have 2,048 full-text indexes for a partitioned table.
4. **Correct Answers: A and C**
   - **A. Correct.** To enable FILESTREAM, you must edit the properties of the SQL Server service in SQL Server Configuration Manager.
   - **B. Incorrect.** You do not have to edit the properties of the Analysis Services service in SQL Server Configuration Manager to enable FILESTREAM.

C. **Correct.** You must run the sp_configure filestream_access_level, X, with a non-zero value of X to enable FILESTREAM.

D. **Incorrect.** Running sp_configure filestream_access_level, 0, disables FILESTREAM.

Lesson 2

1. **Correct Answer: A**

   A. **Correct.** You can add a filegroup to an existing database by using the ALTER DATABASE statement.

   B. **Incorrect.** The CREATE DATABASE statement enables you to create a new database but not to alter an existing database.

   C. **Incorrect.** The ALTER TABLE statement enables you to modify a table but not a database.

   D. **Incorrect.** The CREATE TABLE statement enables you to create a table but not to modify a database.

2. **Correct Answer: B**

   A. **Incorrect.** Implementing page-level compression will provide a greater reduction in the amount of space used than implementing row-level compression will, given the properties of the table.

   B. **Correct.** Unless the table data is unique, page-level compression provides the greatest compression but has a cost in CPU usage.

   C. **Incorrect.** In this case, compressing the index using row compression will not provide the space savings that compressing the entire table will.

   D. **Incorrect.** In this case, compressing the index using page compression will not provide the space savings that compressing the entire table will.

3. **Correct Answers: A, B, and C**

   A. **Correct.** You must create a master encryption key prior to enabling TDE on a database.

   B. **Correct.** You must create a certificate prior to enabling TDE on a database.

   C. **Correct.** You must create a database encryption key prior to enabling TDE on a database.

   D. **Incorrect.** You do not need to enable page-level compression prior to enabling TDE on a database.

4. **Correct Answer: B**

   A. **Incorrect.** DBCC CHECKFILEGROUP verifies the allocation and structural integrity of indexed views and tables for a specific filegroup but not for the entire database.

   B. **Correct.** DBCC CHECKDB checks the physical and logical integrity of all objects within a specific database.

C. **Incorrect.** DBCC SQLPERF provides transaction log space statistics.

D. **Incorrect.** DBCC SHRINKDATABASE shrinks the size of all data and log files for the specified database.

Case Scenario 1

1. To enable FILESTREAM on the default instance of server SYDNEY-DB, you must perform the following general steps:
   A. Edit the properties of the SQL Server Service to enable FILESTREAM.
   B. Run the sp_configure filestream_access_level, X, stored procedure, where X is 1 or 2.
   C. Restart SQL Server Services.
   D. Create a FILESTREAM filegroup.
   E. Add a file to the FILESTREAM filegroup.
2. After FILESTREAM is enabled on the default instance of server SYDNEY-DB, you must take the following general steps to deploy a FileTable:
   A. Enable Non-Transactional Access at the database level.
   B. Specify a directory for FileTables at the database level.
   C. Create a table as a FileTable.

Case Scenario 2

1. Use the following query to create a master encryption key with the password **P@ssw0rd**:

```
USE master;
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'P@ssw0rd';
GO
```

2. Use the following query to create the certificate ServerCertA with the subject name **'Server Certificate A':**

```
CREATE CERTIFICATE ServerCertA WITH SUBJECT = 'Server Certificate A';
GO
```

3. Use the following query to create a DEK for database ContosoCars2012 by using the AES_128 encryption algorithm and certificate ServerCertA:

```
USE ContosoCars2012;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_128
ENCRYPTION BY SERVER CERTIFICATE ServerCertA;
GO
```

4. Use the following statement to encrypt the ContosoCars2012 database:

```
ALTER DATABASE ContosoCars2012
SET ENCRYPTION ON;
GO
```

5. Use the following statement to back up the server certificate to a file named CertExport with a private key file named PrivateKey with the password **P@ssw0rd**:

```
BACKUP CERTIFICATE ServerCertA
TO FILE = 'CertExport'
WITH PRIVATE KEY
(
    FILE = 'PrivateKeyFile',
    ENCRYPTION BY PASSWORD = 'P@ssw0rd'
);
GO
```

CHAPTER 8

Clustering and AlwaysOn

Exam objectives in this chapter:

- Implement a SQL Server clustered instance.
- Implement AlwaysOn.

Failover clustering instances and AlwaysOn Availability Groups are two strategies for making Microsoft SQL Server 2012 databases highly available. Failover clustering is a more traditional approach to ensuring that a database remains available in the event of server failure. If you are planning to deploy a failover cluster instance, you must first deploy the cluster and then install SQL Server 2012 by using a method that differs from a traditional installation.

AlwaysOn Availability Groups are a technology new in SQL Server 2012 that also rely on failover clustering technologies. AlwaysOn Availability Groups are a replacement technology for database mirroring and have the benefit of allowing clients read-only access to the secondary replica. In this chapter, you learn about how to deploy both of these SQL Server 2012 high-availability technologies.

Lessons in this chapter:

Before You Begin

To complete the practice exercises in this chapter, make sure that you have:

- Completed the setup tasks for installing computers DC, SQL-A, SQL-B, and SQL-CORE as outlined in the introduction of this book.
- Completed the setup tasks outlined in the end-of-lesson practice exercises in Chapter 1, "Planning and Installing SQL Server 2012," through Chapter 7, "Mirroring and Replication."
- Deployed two new servers, named SQL-C and SQL-D, in the CONTOSO domain. Instructions for configuring these servers are outlined in the introduction of this book.

No additional configuration is required for this chapter.

Lesson 1: Clustering SQL Server 2012

A SQL Server 2012 failover cluster instance is a special deployment of SQL Server 2012 that stores database files on a shared storage device. If the server that hosts one Database Engine instance fails, another Database Engine instance in the failover cluster takes control of the database files and seamlessly continues to service client requests.

After this lesson, you will be able to:

- Prepare the host operating system for the installation of a failover cluster instance of SQL Server 2012.
- Deploy a failover cluster instance.
- Manage multiple instances on a cluster.
- Deploy multi-subnet failover clusters.
- Troubleshoot failover clusters.

Estimated lesson time: 60 minutes

Fulfilling Edition Prerequisites

You can deploy failover cluster instances only on specific editions of the host operating system and SQL Server. When planning the deployment of a failover cluster instance, remember that:

- SQL Server 2012 Enterprise edition supports up to 16 cluster nodes. This edition of SQL Server is the only one that you can deploy in a production environment that supports multi-subnet failover clustering.
- SQL Server 2012 Business Intelligence edition supports a two-node maximum for failover clusters.
- SQL Server 2012 Standard edition supports a two-node maximum.
- Windows Server 2008 R2 Enterprise and Datacenter editions support failover clustering. These editions also support multi-subnet failover clustering.
- Windows Server 2008 Enterprise and Datacenter editions support failover clustering but do not support multi-subnet failover clustering.

You must have a functional Windows Server failover cluster prior to deploying SQL Server as a failover cluster instance. Only then can you install SQL Server as a failover cluster instance.

MORE INFO **WINDOWS SERVER FAILOVER CLUSTERING WITH SQL SERVER 2012**

You can learn more about Windows Server failover clustering with SQL Server 2012 at *http://msdn.microsoft.com/en-us/library/hh270278.aspx*.

Windows Server 2008 R2 as Shared Storage

Except in the case of multi-subnet failover clusters, SQL Server failover cluster instances require shared storage to host the database and log files. In production situations, you use a dedicated storage area network (SAN) device for this task. If you are using a fiber channel SAN, you use vendor software to make the connection between Microsoft Windows Server 2008 or Windows Server 2008 R2 and the SAN. If you are using iSCSI, you can use the vendor-supplied software or the iSCSI initiator that is included with the server operating system. Connecting by using the iSCSI initiator is covered in the next section, "Connecting to the SAN with iSCSI Initiator."

If you don't have access to a SAN, you can use Windows Storage Server 2008 R2 as an iSCSI target by installing the iSCSI Software Target, which you can download from the Microsoft website. You can use this software to simulate an iSCSI storage device on a SAN when running virtual machines within a Hyper-V environment without actually having to connect to a traditionally deployed SAN.

MORE INFO **DOWNLOAD iSCSI SOFTWARE TARGET**

You can download the iSCSI Software Target at *http://www.microsoft.com/download/en/details.aspx?id=19867*.

After you have installed the iSCSI Software Target on a computer running Windows Server 2008 R2, you must perform several steps to configure the server so other computers can connect and use a specially configured virtual hard disk file as a SAN storage device.

NOTE **iSCSI TARGET**

Configuring the iSCSI target is unlikely to be covered directly on the 70-462 exam, but this is a necessary step to complete the clustering-related practice exercises in this chapter.

The first step you must take to configure the iSCSI Software Target is to configure Windows Firewall with Advanced Security rules on the computer on which you've installed the iSCSI Software Target. You can do this by running the following commands from an elevated command prompt:

```
netsh advfirewall firewall add rule name="Microsoft iSCSI Software Target Service-
TCP-3260" dir=in action=allow protocol=TCP localport=3260

netsh advfirewall firewall add rule name="Microsoft iSCSI Software Target Service-
TCP-135" dir=in action=allow protocol=TCP localport=135

netsh advfirewall firewall add rule name="Microsoft iSCSI Software Target Service-
UDP-138" dir=in action=allow protocol=UDP localport=138

netsh advfirewall firewall add rule name="Microsoft iSCSI Software Target Service"
dir=in action=allow program="%SystemRoot%\System32\WinTarget.exe" enable=yes

netsh advfirewall firewall add rule name="Microsoft iSCSI Software Target Service Status
Proxy" dir=in action=allow program="%SystemRoot%\System32\WTStatusProxy.exe" enable=yes
```

After you have configured the computer that will function as the iSCSI target with the appropriate firewall rules, you can configure the iSCSI Software Target application by performing the following steps:

1. Open the iSCSI Software Target application from the Administrative Tools menu.
2. Right-click the iSCSI Targets node, shown in Figure 8-1, and choose Create iSCSI Target. Click Next.

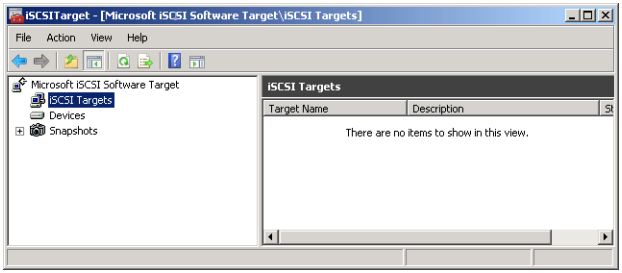

FIGURE 8-1 iSCSI Target console

3. On the iSCSI Target Identification page, enter a name for the target.
4. On the iSCSI Initiators Identifiers page, click Advanced. On the Advanced Identifiers page, click Add.
5. On the Add/Edit Identifiers page, enter the IP address or fully qualified domain name (FQDN) of the hosts that will be accessing the iSCSI target from the network. The wizard presents a warning when adding multiple initiators. Figure 8-2 shows sql-c.contoso.com and sql-d.contoso.com configured as identifiers. Return to the iSCSI Initiators Identifiers page by clicking OK; click Next and then click Finish.

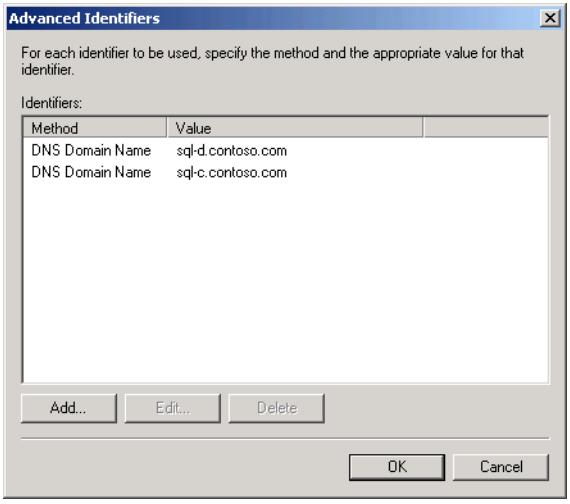

FIGURE 8-2 Advanced identifiers

To create an iSCSI logical unit number (LUN), you create a virtual hard disk (VHD) and make it available. Remember that you must provide a LUN to store quorum information and a LUN to store database files. To provide a LUN by using the iSCSI Software Target, perform the following steps:

1. In the iSCSI Software Target console, right-click Device and choose Create Virtual Disk.
2. On the File page of the Create Virtual Disk Wizard, specify the path to a VHD file that will serve as the storage device for SAN client, for example, d:\SAN\disk-one.vhd.
3. On the Access page, shown in Figure 8-3, click Add to add the iSCSI targets to allow connection to the virtual disk over the network. Click Next and then click Finish.

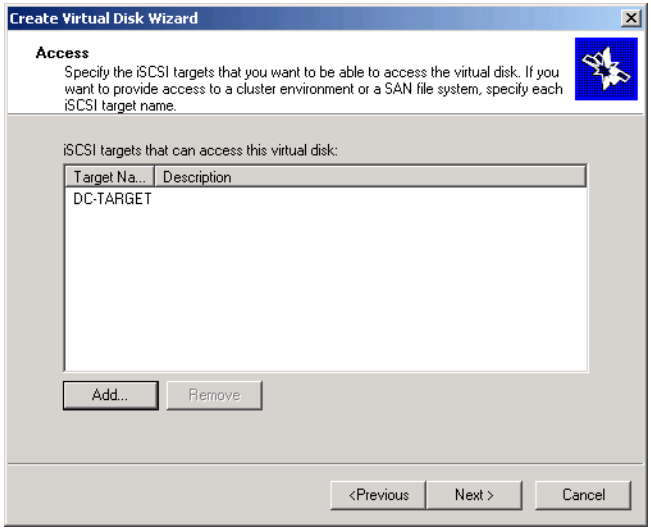

FIGURE 8-3 Access iSCSI target

NOTE **iSCSI SOFTWARE TARGET**

The iSCSI Software Target is included in Windows Server 2012 and does not require a separate download.

Connecting to the SAN by Using iSCSI Initiator

iSCSI Initiator is a component built into the Windows Server 2008 R2 and Windows 7 operating systems that you can use to connect to an iSCSI LUN by using an iSCSI target. When preparing two servers that will function as cluster nodes in a SQL Server 2012 failover cluster, you can configure each server to connect to the same iSCSI LUN for the purposes of shared storage. To connect to an iSCSI LUN by using an iSCSI initiator, perform the following steps:

1. Open iSCSI Initiator from the Administrative Tools menu. If prompted to configure the iSCSI service to start automatically, click Yes.

2. On the Targets tab of the iSCSI Initiator properties, enter the IP address or FQDN of the iSCSI target and click Quick Connect. Verify that the discovered target is correct, as shown in Figure 8-4, and click Done.

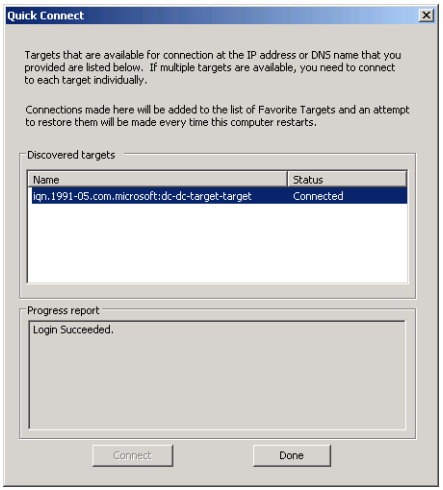

FIGURE 8-4 Discovered target

3. On the Volumes And Devices tab, click Auto Configure and then click OK.
4. Verify that the volumes are available to be brought online and formatted in the Disk Management node of the Server Manager console.

MORE INFO **BEFORE DEPLOYING A FAILOVER CLUSTER INSTANCE**

You can learn more about the steps you must take before deploying a SQL Server failover cluster instance at *http://msdn.microsoft.com/en-us/library/ms189910(SQL.110).aspx*.

Creating a Windows Server 2008 R2 Failover Cluster

The first step in creating a Windows Server 2008 R2 failover cluster to host a SQL Server 2012 failover cluster is to install the Failover Clustering feature. You can do this through the Server Manager console, as shown in Figure 8-5, or by using the following Windows PowerShell command when the ServerManager module is loaded:

```
Add-WindowsFeature Failover-Clustering
```

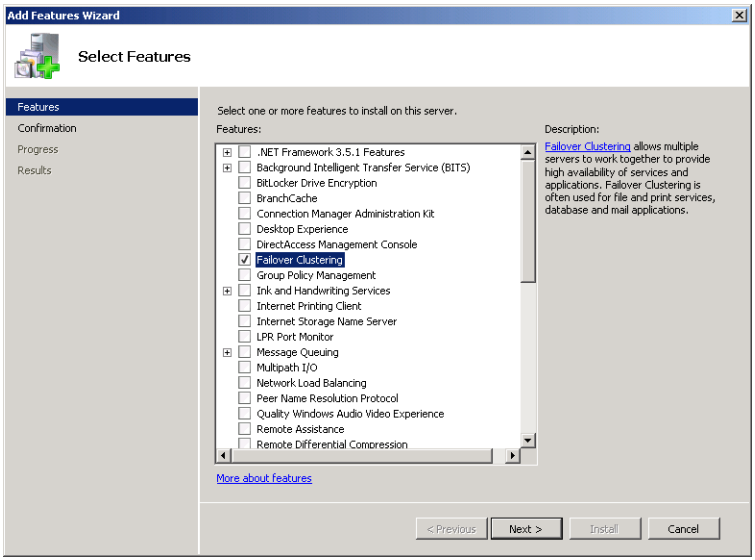

FIGURE 8-5 Installing the Failover Clustering feature

To configure a failover cluster, perform the following steps:

1. When you have connected each potential node to the shared storage device and installed the Failover Clustering feature, open the Failover Cluster Manager from the Administrative Tools menu.
2. In the Failover Cluster Manager console, click Create A Cluster in the Actions menu.
3. On the Select Servers page, shown in Figure 8-6, enter the names of the nodes that will participate in the cluster.

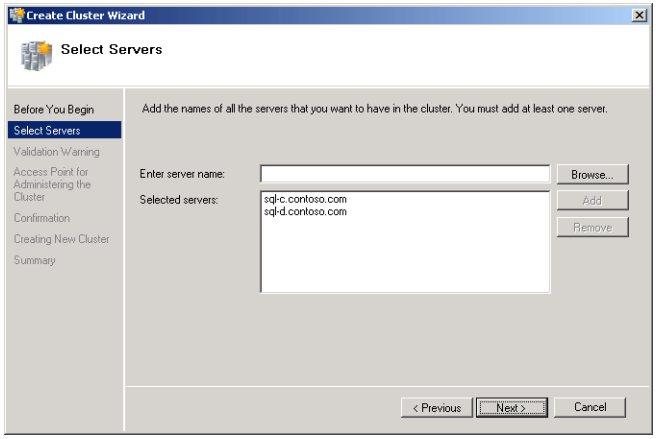

FIGURE 8-6 Selecting cluster nodes

4. Choose whether to perform validation tests.

MORE INFO **VALIDATION TESTS**

Although validation tests are necessary only if you want Microsoft to support the cluster configuration, you should use the tests to identify any potential deviation from best practice.

5. On the Access Point For Administering The Cluster page, enter a name and IP address of the cluster, as shown in Figure 8-7.

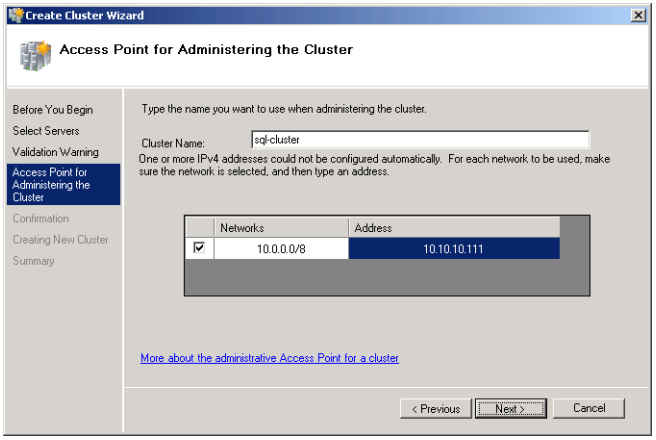

FIGURE 8-7 Cluster administration point

6. On the Confirmation page, verify the settings and click Next to have the wizard create the cluster.

Quick Check

- **Which editions of SQL Server 2012 support more than two-node failover clusters?**

Quick Check Answer

- **Only SQL Server 2012 Enterprise edition supports more than two-node failover clusters.**

Installing a SQL Server Failover Cluster

Installing a SQL Server failover cluster involves performing two installation steps from SQL Server Installation Center. You must first run the Advanced Cluster Preparation Wizard. When this first wizard is complete, you must then run the Advanced Cluster Completion Wizard. You might need to restart the host server between running the first and the second wizards.

If you want to support protocol encryption on a failover cluster instance, you must install a certificate that uses the instance name as a fully qualified domain name on each of the nodes that will host the failover cluster instance prior to running the Advanced Cluster Preparation Wizard.

To prepare a SQL Server failover cluster, perform the following steps:

1. Ensure that the Microsoft .NET Framework 3.5.1 feature is installed.
2. On the first node in the cluster, run setup.exe from the installation media.
3. In the Advanced area of SQL Server Installation Center, shown in Figure 8-8, click Advanced Cluster Preparation.

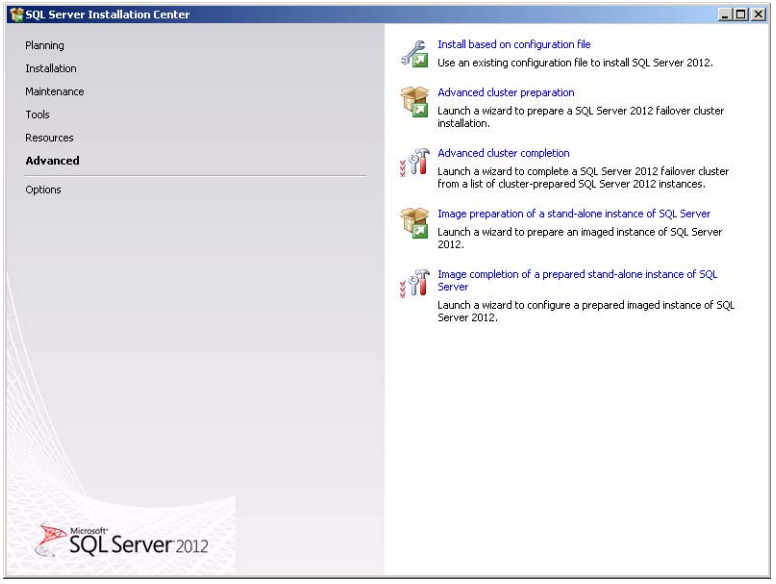

FIGURE 8-8 Advanced cluster preparation

4. On the Product Key page, enter the product key or specify that you use the Evaluation edition. On the License Terms page, select I Accept The License Terms, install any necessary updates, and review the Setup Support Rules warnings.
5. On the Feature Selection page, choose which SQL Server features you want to install on the failover cluster.
6. On the Instance Configuration page, choose the properties of the instance.
7. Review the Disk Space Requirements.
8. On the Server Configuration page, specify a specially configured domain account to be used for the Service Accounts.
9. Review the Error Reporting page.
10. On the Ready To Install page, shown in Figure 8-9, click Install.

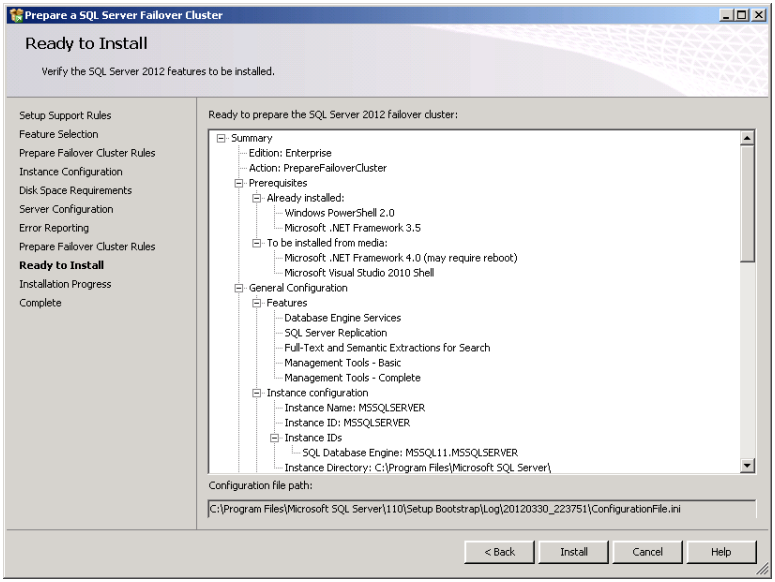

FIGURE 8-9 Preparing failover cluster installation

11. When the process completes, you might need to restart the server.
12. Repeat this process on each node that will participate in the cluster.

Complete the Installation

When you have completed the advanced cluster preparation process on each node that will participate in the failover cluster, return to the cluster node that has ownership of the shared disk and perform the following steps:

1. On the Advanced page of SQL Server Installation Center, click Advanced Cluster Completion. After the setup support rules have run, click OK and then click Next.
2. On the Cluster Node Configuration page, shown in Figure 8-10, specify the SQL Server Instance Name and the SQL Server Network Name that will identify the failover cluster on the network. This network name must be different from any preexisting cluster resource name. Click Next.

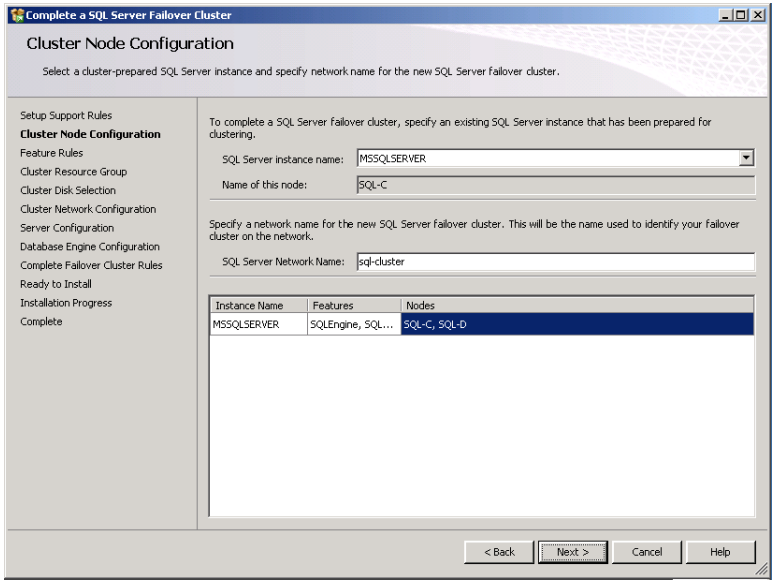

FIGURE 8-10 Cluster node configuration

3. On the Cluster Resource Group page, enter a new name for a new Cluster Resource Group.
4. On the Cluster Disk Selection page, shown in Figure 8-11, specify the disk that will be used as the default drive for databases.

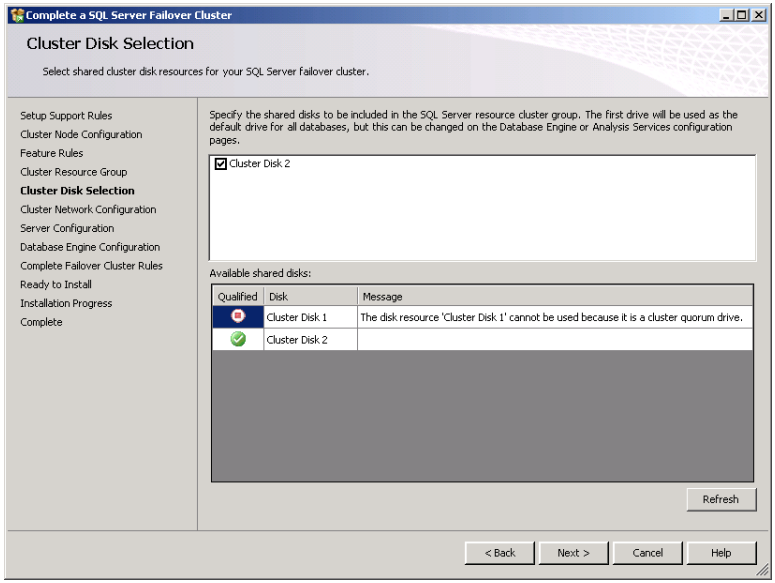

FIGURE 8-11 Cluster disk selection

5. On the Cluster Network Configuration page, specify an IP address for the cluster resource.
6. On the Server Configuration page, verify that the correct collation is set and then click Next.
7. On the Database Engine Configuration page, specify Authentication Mode and the SQL Server administrator.
8. On the Ready To Install page, shown in Figure 8-12, verify the configuration settings and then click Install.

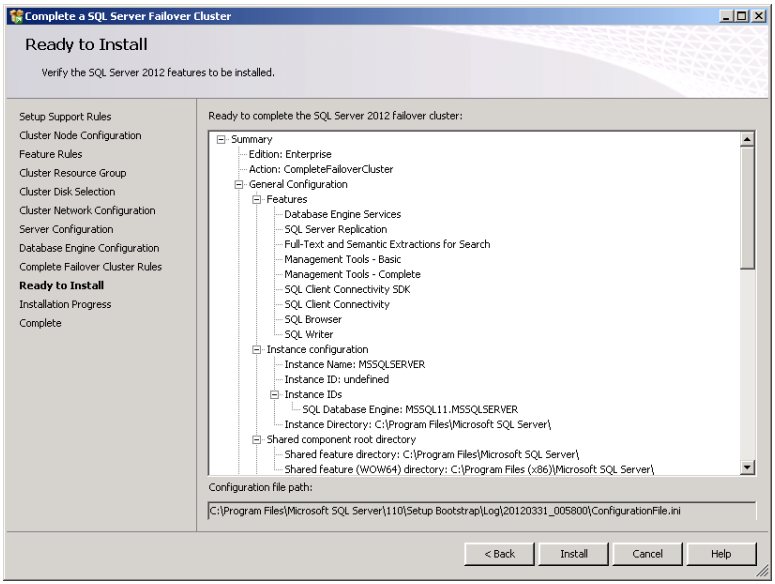

FIGURE 8-12 Final cluster configuration

You must run the Advanced Cluster Completion Wizard only once because this configures all the nodes that you prepared by using the Advanced Cluster Preparation Wizard.

> ***MORE INFO*** **SQL SERVER FAILOVER CLUSTER INSTALLATION**
>
> **You can learn more about the specifics of SQL Server failover cluster instance deployment at *http://msdn.microsoft.com/en-us/library/hh231721(SQL.110).aspx*.**

Multi-Subnet Failover Clustering

Multi-subnet failover clustering is a special configuration where each node in the failover cluster is located on a different TCP/IP subnet. A multi-subnet failover cluster does not use shared storage. When configuring a multi-subnet failover cluster, you must use another solution to replicate data between the instances on separate subnets.

NOTE **STRETCH CLUSTERS**

Stretch cluster **is a term for a geographically dispersed cluster.**

Multi-subnet failover clustering is supported only in production environments in SQL Server 2012 Enterprise edition and Windows Server 2008 R2 Enterprise or Datacenter editions. You cannot deploy multi-subnet failover clustering if the host operating system is running Windows Server 2008. When running the Create A New SQL Server Failover Cluster (Setup) Wizard while configuring a multi-subnet failover cluster, you must ensure that the IP address resource dependency is set to OR.

You can deploy stand-alone instances on servers that also host multi-subnet failover cluster instances. One of the challenges of this configuration is ensuring that communication occurs seamlessly and that no conflicts occur between the multi-subnet failover cluster instance and any stand-alone instances installed on the same host. You can minimize the chance of a conflict occurring by configuring stand-alone instances to use non-default fixed ports and leaving the multi-subnet failover cluster instance to use port 1433.

MORE INFO **MULTI-SUBNET FAILOVER CLUSTERING**

You can learn more about multi-subnet failover clustering at ***http://msdn.microsoft.com/en-us/library/ff878716.aspx*****.**

Performing Manual Failover

You can use the Failover Cluster Manager to perform failover of the cluster resource from one node to another. For example, to perform failover of the SQLCRG resource from SQL-C to SQL-D, perform the following steps:

1. Open Failover Cluster Manager from the Administrative Tools menu.
2. Click the SQLCRG node. On the Actions pane, click Move This Service Or Application To Another Node and then click Move To Node SQL-D.
3. In the Please Confirm Action dialog box, shown in Figure 8-13, click Move SQLCRG To SQL-D.

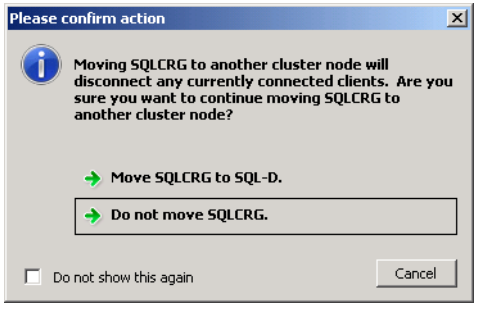

FIGURE 8-13 Moving the cluster resource

4. Verify that SQLCRG comes online on the other node.

EXAM TIP

You can move resources from one node to another node by using the Move-ClusterGroup PowerShell cmdlet when the FailoverClusters module is installed. For example, to move the SQLCRG resource from node SQL-C to node SQL-D, issue the following command:

```
Move-ClusterGroup SQLCRG SQL-D
```

Troubleshooting Failover Clusters

If failover occurs because the primary node suffers irreparable hardware failure, you should perform the following steps:

1. Evict the failed node from the failover cluster instance. You can do this from the Failover Cluster Manager by right-clicking the failed node, choosing Move Actions, and then selecting Evict Node.
2. Verify that the failed node has been successfully evicted from the failover cluster.
3. Replace the hardware that has failed and then use the Failover Cluster Manager console to add the failed node back to the original cluster.
4. After the node has been added to the original cluster, run SQL Server setup to readmit the failed node to the failover cluster instance.

MORE INFO **TROUBLESHOOTING FAILOVER CLUSTERS**

You can learn more about troubleshooting failover clusters at *http://msdn.microsoft.com/en-us/library/ms189117(SQL.110).aspx*.

Quorum failure is a more complicated situation that is generally caused by persistent communications failure or by the problematic configuration of cluster nodes. Quorum failure must be resolved manually by performing the following steps:

1. Start the Windows Server Failover Cluster by using forced quorum on a single node. You do this by choosing Force Cluster Start from the Actions pane of the Failover Cluster Manager.
2. Start the Windows Server Failover Cluster service on additional nodes that can communicate with the node you started by using forced quorum.
3. Configure a new quorum mode and node vote configuration that reflect the realities of the quorum topology. For example, if two nodes are frequently unavailable due to persistent communication failure, reconfigure the quorum mode and vote assignments to remediate this problem.

MORE INFO **RESOLVING QUORUM FAILURE**

You can learn more about resolving quorum failure at *http://msdn.microsoft.com/en-us/library/hh270277.aspx*.

PRACTICE Building a SQL Server 2012 Failover Cluster

In this practice, you create a Windows Server failover cluster and then deploy a SQL Server failover cluster instance. After you have deployed the failover cluster instance, you perform failover.

EXERCISE 1 Configure iSCSI Volumes for Failover Clustering

In this exercise, you configure iSCSI volumes so they can be used as shared storage in a Windows Server failover cluster. You also configure firewall rules and a service account. To complete this exercise, perform the following steps:

1. Log on to server DC with the Kim_Akers user account.
2. Ensure that the SQL-C and SQL-D computer accounts are included in the SQL Server organizational unit (OU).
3. Download and install the iSCSI Software Target.
4. Create a folder named **C:\SAN**.
5. On DC, configure inbound rules for TCP ports 135 and 3260 and for the inbound rule for the following executables:
   - %systemroot%\System32\WinTarget.exe
   - %systemroot%\System32\WTStatusProxy.exe
6. On DC, create an iSCSI target named **DC-TARGET** and configure it to be accessible to sql-c.contoso.com and sql-d.contoso.com.
7. Create a virtual disk named **c:\SAN\disk-one.vhd**. Set the size to 2 GB. Allow access to the DC-TARGET iSCSI target.
8. Create a virtual disk named **c:\SAN\disk-two.vhd**. Set the size to 10 GB. Allow access to the DC-TARGET iSCSI target.
9. Use the iSCSI initiator to connect to the domain controller as a target on both SQL-C and SQL-D.
10. Using the Disk Management node of the Server Manager console on SQL-C to bring each of the two volumes online, initialize them and create new simple volumes formatted with the NTFS file system.
11. Use the DNS console to create a DNS A record for the address sql-cluster.contoso.com mapped to IP address 10.10.10.111.

12. Use Active Directory Users And Computers to create a user account named **SQL-Cluster** with the password **Pa$$w0rd**.
13. Edit the Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Log On As A Service policy in the SQL-POLICY Group Policy Object (GPO) and grant the SQL-Cluster user account the Log On As A Service right.

EXERCISE 2 Configure a Windows Server 2008 R2 Failover Cluster

In this exercise, you configure a Windows Server 2008 R2 failover cluster by using the shared storage device configured in the previous exercise. To complete this exercise, perform the following steps:

1. When logged on with the Kim_Akers user account, install the Failover Clustering and .NET Framework 3.5.1 features on SQL-C and SQL-D.
2. Run the Create Cluster Wizard from the Failover Cluster Manager console. Configure the failover cluster with the following properties:
   - Cluster Servers: SQL-C and SQL-D
   - Cluster Name: SQL-Cluster
   - Cluster IP Address: 10.10.10.111

EXERCISE 3 SQL Server Failover Cluster Advanced Cluster Preparation

In this exercise, you run the advanced cluster preparation process on the nodes that will participate in the failover cluster instance. To complete this exercise, perform the following steps:

1. Log on to servers SQL-C and SQL-D with the Kim_Akers user account.
2. From SQL Server Installation Center, run Advanced Cluster Preparation on SQL-C and SQL-D with the following options:
   - Install the Database Engine Services, SQL Server Replication, Management Tools - Basic, and Management Tools - Complete features.
   - Use the Default instance with the default settings.
   - Use the CONTOSO\SQL-Cluster account for the SQL Agent and Database Engine service accounts.
3. You might need to restart SQL-C and SQL-D to complete the advanced cluster preparation.

EXERCISE 4 SQL Server Failover Cluster Advanced Cluster Completion

In this exercise, you complete the failover cluster instance installation process. To complete this exercise, perform the following steps:

1. Log on to server SQL-C with the Kim_Akers user account.
2. Verify that SQL-C has control of the two SAN disks that will be used with the cluster.

3. On the Advanced page of SQL Server Installation Center, run Advanced Cluster Completion and provide the following settings:
   - SQL Server network name: SQL2012Cluster
   - Cluster resource group name: SQLCRG
   - Cluster Network Configuration IP address: 10.0.0.120 with subnet mask 255.255.255.0
   - Use Windows Authentication mode; set CONTOSO\Kim_Akers as SQL Server administrator
4. After installation is complete, open the Failover Cluster Manager and verify that the SQLCRG service is online, as shown in Figure 8-14.

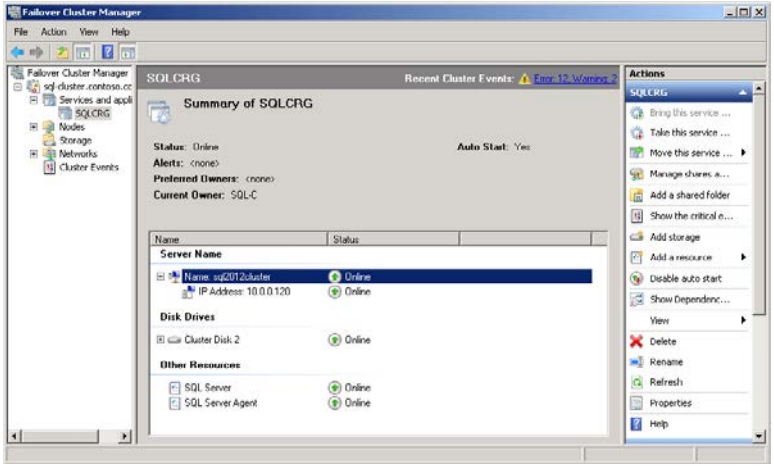

FIGURE 8-14 Verifying cluster configuration

EXERCISE 5 Perform Cluster Failover

In this exercise, you perform failover of the failover cluster instance. To complete this exercise, perform the following steps:

1. Log on to server SQL-C with the Kim_Akers user account.
2. Use Failover Cluster Manager to move the SQLCRG resource from SQL-C to SQL-D.
3. Verify that SQLCRG comes online on the other node and that the current owner is set to SQL-D.
4. Use the appropriate PowerShell cmdlet to move the SQLCRG resource back to SQL-C from SQL-D.

Lesson Summary

- A Windows Server Failover Cluster must be created prior to installing a failover cluster instance.
- Windows Server 2008 Enterprise and Datacenter editions and Windows Server 2008 R2 Enterprise and Datacenter editions can function as host operating systems for failover cluster instances.
- To install a failover cluster instance, first run advanced cluster preparation on all nodes and then run advanced cluster completion on the node that has control of the shared storage device.
- Multi-subnet failover clusters have nodes on separate TCP/IP subnets.
- Use the Failover Cluster Manager console or the Move-ClusterGroup PowerShell cmdlet to perform manual failover.
- In the event of hardware failure, evict the failed node from the cluster and then join it after it is repaired before reinstalling SQL Server.

Lesson Review

Answer the following questions to test your knowledge of the information in this lesson. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the "Answers" section at the end of this chapter.

1. Which of the following operating systems can you use as the host operating system for a SQL Server 2012 multi-subnet failover cluster instance?
   - **A.** Windows 7 Enterprise edition
   - **B.** Windows Server 2008 Enterprise edition
   - **C.** Windows Server 2008 R2 Enterprise edition
   - **D.** Windows Vista Ultimate edition
2. Which of the following Windows PowerShell commands can you use to perform failover of a SQL Server failover cluster instance from one node to another?
   - **A.** Move-ClusterGroup
   - **B.** Move-ClusterResource
   - **C.** Move-ClusteredSharedVolume
   - **D.** Move-ClusterVirtualMachineRole
3. The primary node of a four-node SQL Server failover cluster instance fails due to a hardware failure. Replacement hardware will not arrive for 48 hours. Which of the following steps should you take first to remedy this situation?
   - **A.** Evict the failed node.
   - **B.** Evict the new primary node.

C. Reinstall SQL Server on the failed node.

D. Join the failed node to the cluster.

4. You have configured servers SYD-A and SYD-B to be members of a Windows Server failover cluster. Server SYD-B has control of the shared disk resources. You will deploy SQL Server 2012 as a failover cluster instance on these servers. Which of the following steps must you take to accomplish this goal? (Each correct answer forms part of a complete solution. Choose all that apply.)

   A. Run Advanced Cluster Preparation on SYD-A.

   B. Run Advanced Cluster Preparation on SYD-B.

   C. Run Advanced Cluster Completion on SYD-A.

   D. Run Advanced Cluster Completion on SYD-B.

Lesson 2: AlwaysOn Availability Groups

This lesson covers AlwaysOn Availability Groups, a high-availability feature that is new in SQL Server 2012. In this lesson, you learn about the infrastructure requirements for implementing AlwaysOn and what steps to take to enable AlwaysOn functionality.

After this lesson, you will be able to:

- Configure an instance to support AlwaysOn Availability Groups.
- Create and configure availability groups.
- Add and remove databases from availability groups.
- Perform availability group failover.

Estimated lesson time: 60 minutes

What Are AlwaysOn Availability Groups?

AlwaysOn Availability Groups are an alternative to database mirroring. An *availability group* is a collection of user databases, termed *availability databases*, that can fail over together. Unlike mirroring that is limited to a principal and a mirror database, availability groups support a set of read-write primary databases and up to four sets of secondary databases. Availability groups also enable you to configure one or more sets of secondary databases so that they are accessible for read-only operations.

Failover occurs on a per-replica basis, and all databases in the replica fail over. Database failover is not caused by issues related to individual databases, such as database file or transaction log corruption, but by factors at the instance level, as is the case with normal failover clusters. Availability groups support automatic failover.

Although you must deploy AlwaysOn Availability Groups on an instance that resides on a failover cluster, you usually do not deploy availability groups on a failover cluster instance. Put another way, even though you deploy AlwaysOn with a cluster, you install availability groups on an instance that was deployed by using the typical method outlined in Chapter 1 rather than by using the advanced cluster preparation and advanced cluster completion processes outlined in Lesson 1, "Clustering SQL Server 2012," of this chapter. You can deploy AlwaysOn Availability Groups on a Windows Server failover cluster that does not include a shared storage resource.

MORE INFO **ALWAYSON AVAILABILITY GROUPS**

You can learn more about availability groups at *http://msdn.microsoft.com/en-us/library/hh510230.aspx*.

Meeting Availability Group Prerequisites

For production environments, only SQL Server 2012 Enterprise edition supports AlwaysOn Availability Groups. When planning the deployment of AlwaysOn Availability Groups, the host server must meet the following conditions:

- Host servers cannot be domain controllers.
- Each host server must be a participant node in a Windows Server failover cluster. Failover clustering is supported only on Windows Server 2008 Enterprise and Datacenter editions and Windows Server 2008 R2.
- You must ensure that appropriate hotfixes are applied to the host server operating system.

Although not a requirement, best practice is to ensure all host systems that participate in an availability group can handle identical workloads and to provide host systems with separate network adapters dedicated for availability group traffic. You should also configure a Time To Live (TTL) of 60 seconds on the zone that hosts the DNS records related to the availability group.

If you must support Kerberos authentication with availability groups, you must perform the following extra steps:

- The SQL Server service on each participating instance must use the same domain account.
- You must manually register a Service Principal Name (SPN) for the virtual network name (VNN) of the availability group listener with the domain account used as each instance's SQL Server service account.

These steps are unnecessary if you are using the default NTLM authentication option.

MORE INFO **AVAILABILITY GROUP PREREQUISITES**

You can learn more about availability group prerequisites at *http://msdn.microsoft.com/en-us/library/ff878487.aspx*.

Configuring Availability Modes

AlwaysOn Availability Groups support similar modes to database mirroring. The type of availability mode that is appropriate depends on data loss and transaction latency requirements. You configure availability modes on a per-availability replica basis. AlwaysOn Availability Groups support the following availability modes:

- **Asynchronous-commit mode** This mode is suitable when you must place availability replicas at geographically dispersed locations. When you configure all secondary replicas to use asynchronous-commit mode, the primary will not wait for secondaries to harden the log (write log records to disk) and will run with minimum transaction

latency. If you configure the primary to use asynchronous-commit mode, the transactions for all replicas will be committed asynchronously independently of which mode you've configured on each secondary replica.

- **Synchronous-commit mode** This mode increases transaction latency but minimizes the chance of data loss in the event of automatic failover. When you use this mode, each transaction is applied to the secondary replica before being written to the local log file. The primary verifies that the transaction has been applied to the secondary before entering a SYNCHRONIZED state.

You can configure the availability mode on the Availability Group Properties page, as shown in Figure 8-15. You can also use the ALTER AVAILABILITY GROUP Transact-SQL statement with the AVAILABILITY_MODE option to change the availability mode. For example, to change the availability mode of the SQL-C\AlwaysOn replica to synchronous commit for the AG-Alpha availability group, execute the statement:

```
ALTER AVAILABILITY GROUP AG-ALPHA MODIFY REPLICA ON 'SQL-C\AlwaysOn' WITH
( AVAILABILITY_MODE = SYNCHRONOUS_COMMIT);
```

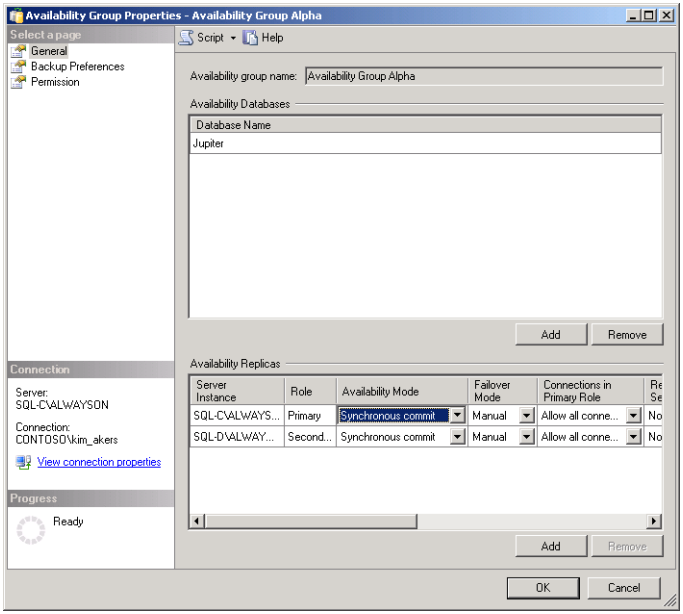

FIGURE 8-15 Synchronous-commit availability mode

> ***MORE INFO*** **AVAILABILITY MODES**
>
> You can learn more about availability modes at *http://msdn.microsoft.com/en-us/library/ff877931.aspx*.

Selecting Failover Modes

Availability groups fail over at the availability-replica level. Failover involves another instance becoming the primary replica, with the original primary replica being demoted to become a secondary replica. AlwaysOn Availability Groups support three forms of failover:

- **Automatic failover** This form of failover occurs without administrator intervention. No data loss occurs during automatic failover. Automatic failover is supported only if the current primary and at least one secondary replica are configured with a failover mode set to AUTOMATIC, and at least one of the secondary replicas set to AUTOMATIC is also synchronized. Automatic failover can occur only if the primary and replica are in synchronous-commit mode, as shown in Figure 8-16.

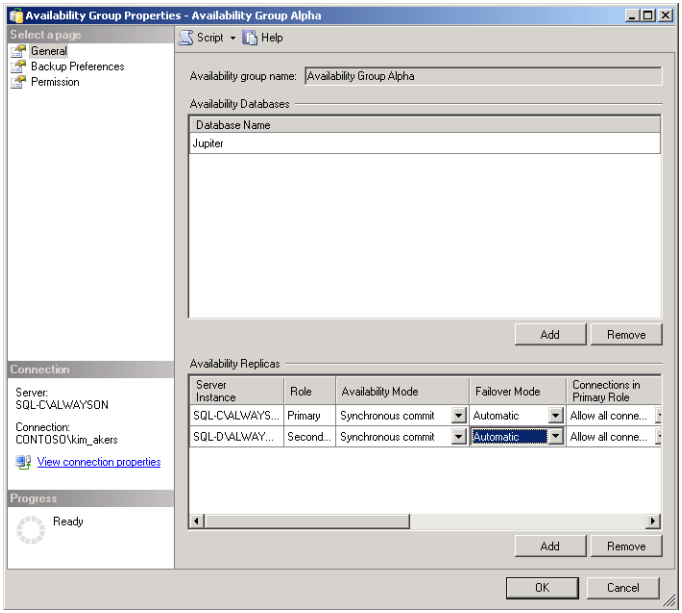

FIGURE 8-16 Automatic failover with synchronous commit

- **Planned manual failover** This form of failover is triggered by an administrator. No data loss occurs during planned manual failover. You perform this type of failover when you must perform a type of maintenance on a host instance that requires the instance or the host server to be taken offline or restarted. Planned manual failover can occur only if at least one of the secondary replicas is in a SYNCHRONIZED state. You can perform planned manual failover only if the primary and replica instances are in synchronous-commit mode.
- **Forced manual failover** This form of failover involves the possibility of data loss. Use forced manual failover when no secondary replica is in the SYNCHRONIZED state or when the primary replica is unavailable. This type of failover is the only type supported

if asynchronous-commit mode is used on the primary, or if the only available replica uses asynchronous-commit mode.

To perform manual failover by using SQL Server Management Studio, perform the following steps:

1. Connect to the server instance that hosts the secondary replica of the availability group that you will make the primary replica.
2. Right-click the availability group and click Failover. This starts the Fail Over Availability Group Wizard.
3. On the Select New Primary Replica page, shown in Figure 8-17, select the instance on which to perform failover and then click Next.

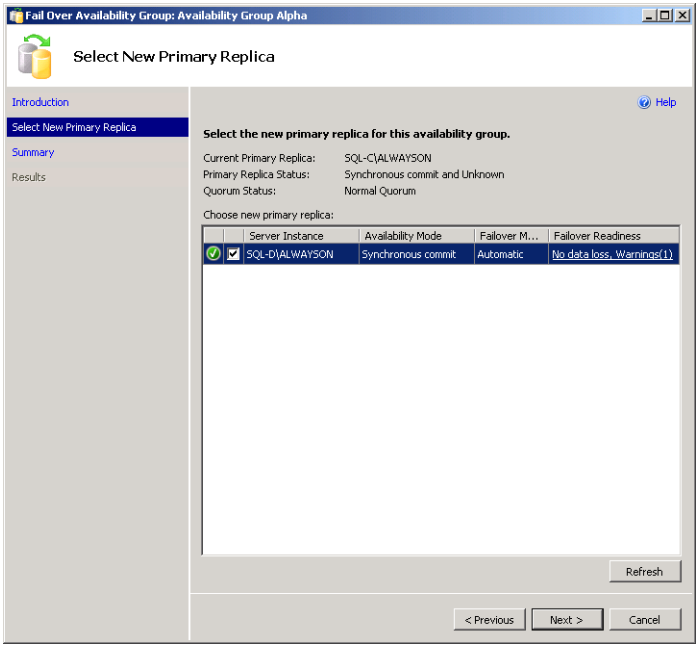

FIGURE 8-17 Manual failover

You can use the ALTER AVAILABILTY GROUP statement with the FAILOVER option on the replica instance that you will make the primary instance. For example, to perform manual failover of the AG-Alpha availability group, execute the following statement:

```
ALTER AVAILABILITY GROUP AG-Alpha FAILOVER;
```

You can use the Switch-SqlAvailabilityGroup PowerShell cmdlet to perform manual failover. For example, to perform manual failover of availability group AG-Alpha to the SQL-D\AlwaysOn instance, execute the command:

```
Switch-SqlAvailabilityGroup -Path SQLSERVER:\SQL\SQL-D\AlwaysOn\AvailabilityGroups\AG-Alpha
```

To perform forced failover by using SQL Server Management Studio, perform the following steps:

1. Connect to the server instance that hosts the secondary replica of the availability group you will make the primary replica.
2. Right-click the availability group and click Failover. This starts the Fail Over Availability Group Wizard.
3. On the Select New Primary Replica page, select the instance on which to perform failover.
4. On the Confirm Potential Data Loss page, shown in Figure 8-18, select Click Here To Confirm Failover With Potential Data Loss and click Next.

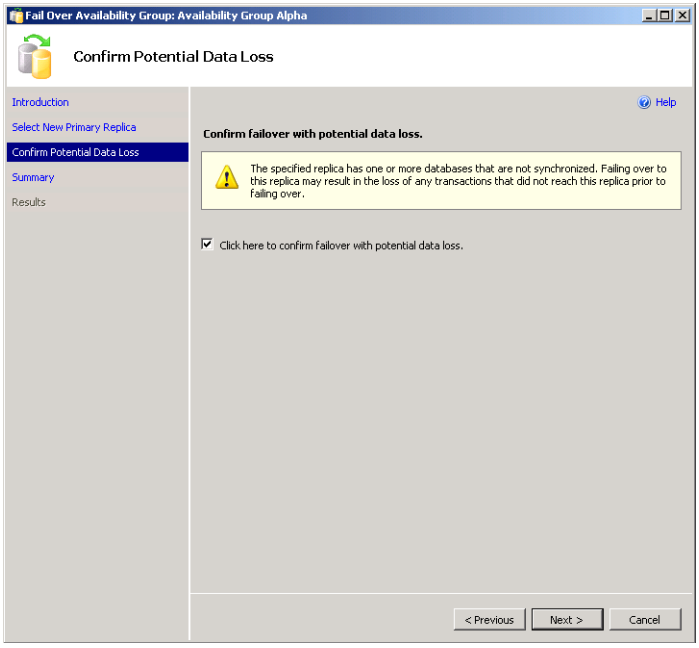

FIGURE 8-18 Confirming potential data loss

You can use the ALTER AVAILABILTY GROUP statement with the FORCE_FAILOVER_ALLOW_DATA_LOSS option on the replica instance that you will make the primary instance to force failover. For example, to force failover of the AG-Alpha availability group, execute the following statement:

```
ALTER AVAILABILITY GROUP AG-Alpha FORCE_FAILOVER_ALLOW_DATA_LOSS;
```

You can use the Switch-SQLAvailabilityGroup PowerShell cmdlet with the AllowDataLoss option to force failover. For example, to force failover of availability group AG-Alpha to the SQL-D\AlwaysOn instance, execute the command:

```
Switch-SqlAvailabilityGroup -Path SQLSERVER:\SQL\SQL-D\AlwaysOn\AvailabilityGroups\AG-
Alpha -AllowDataLoss
```

You can also use the Force option with the preceding PowerShell command if you do not want to be prompted, such as when using the command in a script.

> ***MORE INFO*** **FAILOVER MODES**
>
> **You can learn more about failover modes at *http://msdn.microsoft.com/en-us/library/hh213151.aspx*.**

Configuring Readable Secondary Replicas

Readable secondary replicas can service read-only requests for database access, which enables you to offload read-only workloads from the primary replica. You can configure a secondary replica to be readable from the Availability Group Properties dialog box, as shown in Figure 8-19. There are three options when configuring a readable secondary: No, Yes, and Read-intent only. The difference between Yes and Read-intent is that when you configure Read-intent, only read-only connections are allowed to the secondary databases on the secondary replica. When you configure Yes, all connections are allowed to secondary databases on the secondary replica but only for read access.

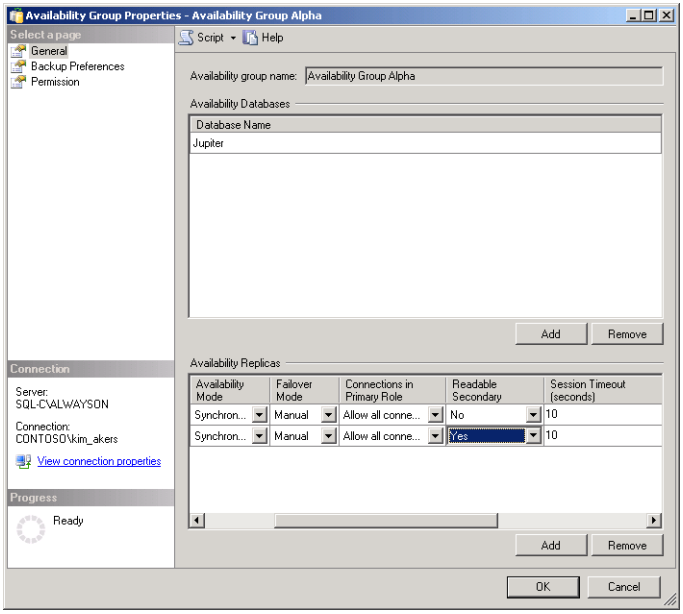

FIGURE 8-19 Readable secondary replicas

You can configure readable secondary properties for a replica by using the ALTER AVAILABILITY GROUP Transact-SQL statement with the SECONDARY_ROLE option.

MORE INFO **READABLE SECONDARY REPLICAS**

You can learn more about readable secondary replicas at *http://msdn.microsoft.com/en-us/library/ff878253.aspx*.

Quick Check

- **Which availability mode is more suitable when replicas are located in geographically dispersed sites?**

Quick Check Answer

- **Asynchronous-commit mode is more suitable for availability replicas distributed over geographically dispersed topologies.**

Deploying AlwaysOn Availability Groups

Even when you have the requisite instances deployed on a Windows Server failover cluster, deploying AlwaysOn Availability Groups involves performing several tasks in order. These tasks are as follows:

- Creating a mirroring endpoint
- Enabling AlwaysOn
- Creating an availability group
- Creating an availability group listener
- Adding a secondary replica

Creating an AlwaysOn Endpoint

Unless you are using a domain-based account for each SQL Server service, you must create a mirroring *endpoint* prior to creating an AlwaysOn Availability Group. If you are using a domain-based account for all SQL Server services that will participate in the availability group, the Database Engine can create the appropriate mirroring endpoint automatically as part of the availability group creation process.

Prior to creating the endpoint, check whether there is an existing endpoint on the instance because you can have only one mirroring endpoint on an instance. You can check whether there are any mirroring endpoints on an instance by querying the sys.database_mirroring_endpoints catalog view.

NOTE **MIRRORING ENDPOINTS**

You can use only mirroring endpoints that use Windows Authentication if you are using a domain account form of the SQL Server service account. If you are using a local account with the SQL Server service, you should configure certificate-based authentication. You learned about creating mirroring endpoints in Chapter 7.

You can create an endpoint from the SQL Server PowerShell module by using the New-SqlHadrEndpoint cmdlet. For example, to create an endpoint named AlwaysOnEndpoint that uses TCP port 7028 on instance SQL-A\ALTERNATE, issue the command:

```
$endpoint = New-SqlHadrEndpoint AlwaysOnEndpoint -Port 7028 -Path SQLSERVER:\SQL\SQL-A\
ALTERNATE
```

After an endpoint has been created, you must start that endpoint. You can do so by using the Set-SqlHadrEndpoint cmdlet. For example, to start the endpoint created in the previous example, issue the command:

```
Set-SqlHadrEndpoint -InputObject $endpoint -State "Started"
```

MORE INFO **CREATING AN ALWAYSON ENDPOINT**

You can learn more about creating mirroring endpoints for AlwaysOn at *http://msdn.microsoft.com/en-us/library/hh510204.aspx*.

Enabling AlwaysOn Availability Groups

Before you can create an AlwaysOn Availability Group, you must enable the AlwaysOn Availability Groups functionality at the instance level. To enable AlwaysOn Availability Groups on an instance, perform the following steps:

1. In SQL Server Configuration Manager, navigate to the SQL Server Services node.
2. Right-click the SQL Server service related to the instance on which you want to enable AlwaysOn Availability Groups.
3. On the AlwaysOn High Availability tab, select Enable AlwaysOn Availability Groups, as shown in Figure 8-20.

   This tab should also display the name of the failover cluster to which the node belongs.

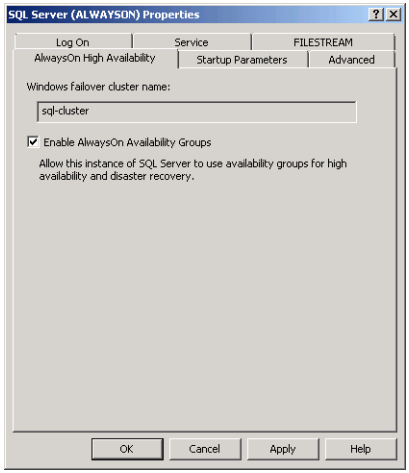

FIGURE 8-20 Enabling AlwaysOn

4. You must now restart the SQL Server service before AlwaysOn is enabled. When enabling AlwaysOn Availability Groups, you should enable only one instance at a time. You should then wait until the SQL Server service has restarted before enabling AlwaysOn on other instances that will participate in the availability group.

You can also enable AlwaysOn by using SQL Server PowerShell with the Enable-SQLAlwaysOn cmdlet. For example, to enable AlwaysOn on the ALTERNATE instance on server SQL-B, issue the following command:

```
Enable-SqlAlwaysOn -Path SQLSERVER:\SQL\SQL-B\ALTERNATE
```

If you choose to disable AlwaysOn on a Database Engine instance, either by using the Disable-SqlAlwaysOn PowerShell cmdlet or by using SQL Server Configuration Manager, you must restart the associated SQL Server service.

MORE INFO **ENABLING ALWAYSON**

You can learn more about enabling AlwaysOn at *http://msdn.microsoft.com/en-us/library/ff878259.aspx*.

Creating an Availability Group

After AlwaysOn is enabled at the Database Engine instance level, you can create availability groups. To create an availability group by using SQL Server Management Studio, perform the following steps:

1. In SQL Server Management Studio, on the instance that hosts the primary replica, expand the AlwaysOn High Availability node.
2. Right-click Availability Groups and click New Availability Group Wizard.

3. On the Specify Availability Group Name page, provide a name for the availability group.
4. On the Select User Databases For The Availability Group page, shown in Figure 8-21, select the databases you will add to the availability group. You cannot create an availability group by using this wizard unless you can add at least one database. This page also informs you of whether the database meets the availability group's prerequisites or must be backed up before it can be added.

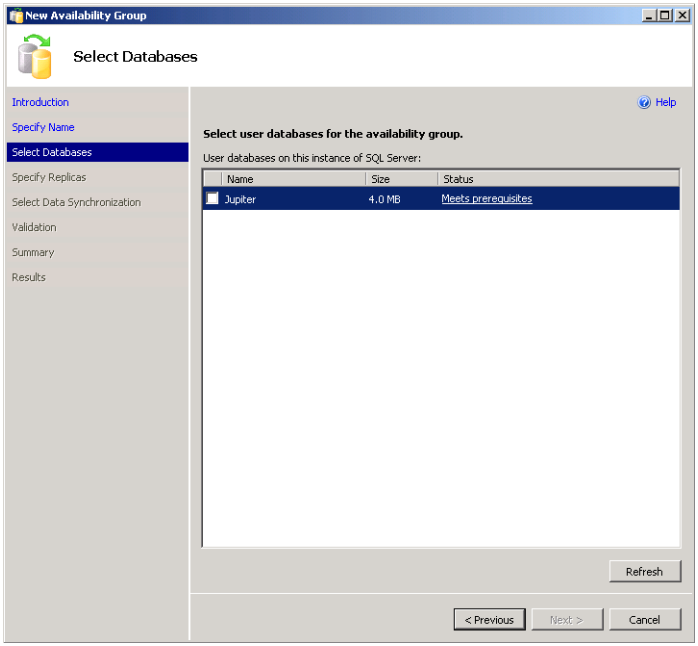

FIGURE 8-21 Adding a database to an availability group

5. On the Specify Replicas page, shown in Figure 8-22, click Add Replica. In the Connect To Server dialog box, specify the credentials you use to connect. Add the instances that will function as replicas. You can also use this page of the wizard to configure an availability group listener.

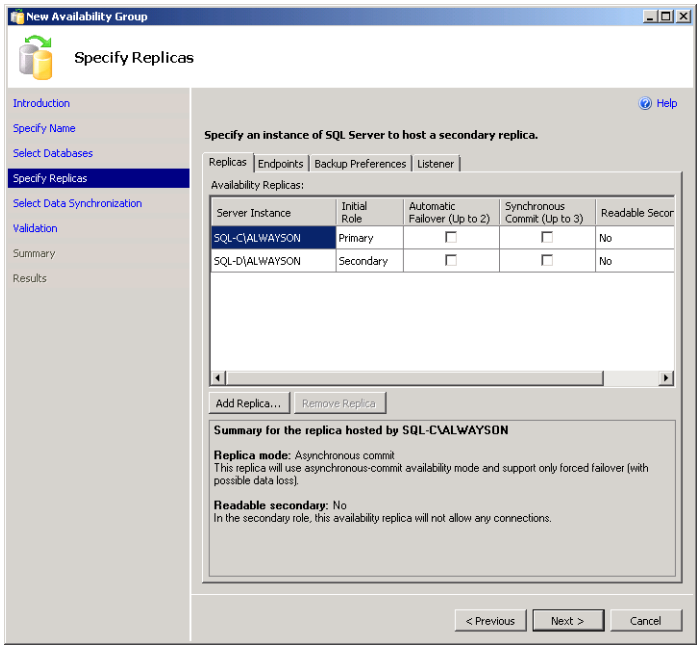

FIGURE 8-22 Specifying replicas

6. On the Select Initial Data Synchronization page, specify the location of a network share that allows read/write access to the SQL Server service account of all replicas.
7. On the Validation page, verify that all processes except Checking The Listener Configuration are competed successfully. You learn how to create an availability group listener in the "Creating or Adding an Availability Group Listener" section in this chapter.
8. Review the summary and complete the wizard.

You cannot use the New Availability Group Wizard or Add Database To Availability Group Wizard to add a database to an availability group if that database is encrypted or contains a Database Encryption Key. You also cannot use the New Availability Group Wizard to add replicas that use different paths for database and log files. You must add these replicas manually. You learn how to perform this task in the "Adding Secondary Replicas" section in this chapter.

MORE INFO **NEW AVAILABILITY GROUP WIZARD**

You can learn more about the New Availability Group Wizard at *http://msdn.microsoft.com/en-us/library/hh403415.aspx*.

You can use the CREATE AVAILABILITY GROUP Transact-SQL statement to create an availability group. For example, to create an availability group with the following properties:

- Name: AG-BETA
- Database: Saturn
- Replica instances: SQL1.contoso.com\newinstance, SQL2.contoso.com\newinstance
- Endpoint TCP port: 7030
- Failover mode: Manual
- Availability mode: Asynchronous

execute the following Transact-SQL code:

```
CREATE AVAILABILITY GROUP AG-BETA
        FOR
                 DATABASE Saturn
        REPLICA ON
        'SQL1\newinstance' WITH
                 (
                 ENDPOINT_URL = 'TCP://sql1.contoso.com:7030',
                 AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT,
                 FAILOVER_MODE = MANUAL
        ),
        'SQL2\newinstance' WITH
                  (
                 ENDPOINT_URL = 'TCP://sql2.contoso.com:7030',
                 AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT,
                 FAILOVER_MODE = MANUAL
        );
GO
```

MORE INFO **CREATING AN AVAILABILITY GROUP BY USING TRANSACT-SQL**

You can learn more about creating an availability group by using Transact-SQL at *http://msdn.microsoft.com/en-us/library/ff878307.aspx*.

Creating or Adding an Availability Group Listener

An *availability group listener* is a network connectivity endpoint for an availability group. Clients connect to the listener, which in turn connects them to the availability group's primary instance. You can create one listener per availability group by using SQL Server Management Studio. If you need more than one listener for an availability group, it is possible to create additional listeners by using Windows PowerShell or the Failover Cluster Manager console.

To create an availability group listener, you must be connected to the Database Engine instance that hosts the primary replica. To create an availability group listener for an existing availability group by using SQL Server Management Studio, perform the following steps:

1. In SQL Server Management Studio, navigate to the AlwaysOn High Availability node and then expand the Availability Groups node. Right-click the availability group for which you will create a listener and click Add Listener.

2. On the New Availability Group Listener page, shown in Figure 8-23, specify a Listener DNS Name and a TCP Port. In the Network Mode box, select either DHCP or Static IP. If using a static IP, specify the static IP address.

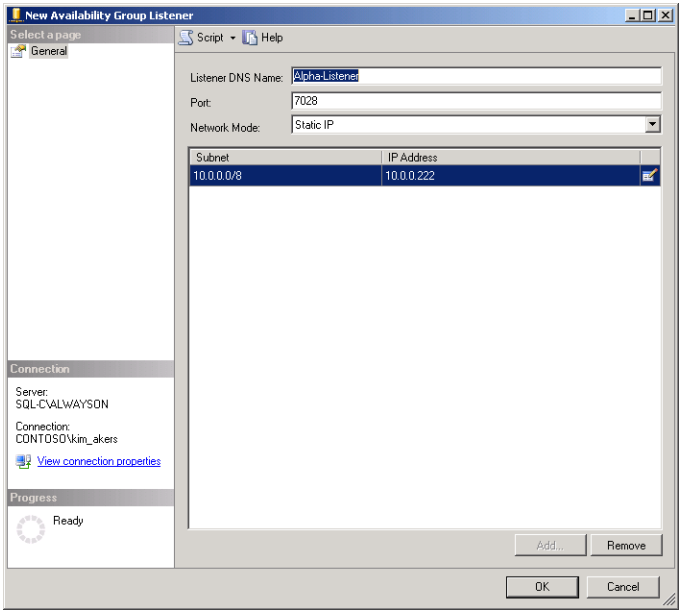

FIGURE 8-23 Availability group listener

You can add a listener to an existing availability group by using the ALTER AVAILABILITY GROUP Transact-SQL statement. For example, to add a listener named Beta-Listener to the AG-Alpha availability group that uses IP address 10.0.0.222, subnet mask 255.0.0.0, and port 7028, execute the statement:

```
ALTER AVAILABILITY GROUP [AG-Alpha]
      ADD LISTENER 'Beta-Listener' (with IP (('10.0.0.222','255.0.0.0')), PORT=7028);
GO
```

You can create an availability group listener by using the New-SqlAvailabilityGroupListener cmdlet. For example, to create a new availability group listener on instance SQL-C\AlwaysOn named Gamma-Listener to the AG-Gamma availability group that uses IP address 10.0.0.224, subnet mask 255.0.0.0, and port 7030, use the command:

```
New-SqlAvailabilityGroupListener -Name Gamma-Listener -StaticIP '10.0.0.224/255.0.0.0'
-Port 7030 -Path SQLSERVER:\SQL\SQL-C\ALWAYSON\AvailabilityGroups\AG-Gamma
```

MORE INFO **AVAILABILITY GROUP LISTENERS**

You can learn more about availability group listeners at *http://msdn.microsoft.com/en-us/library/hh213080.aspx*.

Adding Secondary Replicas

You can add secondary replicas to an existing availability group under the following conditions:

- The availability group has fewer than four secondary replicas.
- The primary replica of the availability group is online.
- You are connected to the Database Engine instance that will host the secondary replica.
- The Database Engine instance that will host the secondary replica can connect to the mirroring endpoint on the primary replica.
- You have enabled AlwaysOn Availability Groups on the Database Engine instance that will host the secondary replica.

To join a secondary replica to an availability group by using SQL Server Management Studio, perform the following steps:

1. On the Database Engine instance that hosts the secondary replica, right-click the secondary replica under the AlwaysOn High Availability\Availability Groups node and click Join To Availability Group.
2. In the Join Replica To Availability Group dialog box, verify the details and click OK.

You can use the ALTER AVAILABILITY GROUP Transact-SQL statement to join a secondary replica to an availability group. For example, to join the AG-Delta availability group, execute the following Transact-SQL statement on the Database Engine instance that hosts the secondary replica:

```
ALTER AVAILABILITY GROUP AG-Delta JOIN;
```

You can also use the Join-SqlAvailabilityGroup cmdlet to join a secondary replica to an availability group. For example, to join the SQL-E\AlwaysOn instance to the AG-Delta availability group, execute the command:

```
Join-SqlAvailabilityGroup -Path SERVER:\SQL\SQL-E\AlwaysOn -Name 'AG-Delta'
```

> ***MORE INFO*** **SECONDARY REPLICAS**
>
> **You can learn more about adding secondary replicas to availability groups at *http://msdn.microsoft.com/en-us/library/ff878473.aspx*.**

Using Availability Groups on Failover Cluster Instances

Although you must deploy availability groups on a host server that is a member of a Windows Server failover cluster, the instance on which you deploy availability groups is not usually a failover cluster instance. You can use availability groups with SQL Server failover cluster

instances, but you cannot use all availability group functionality. The following restrictions apply in this scenario:

- Only one failover cluster instance partner can host a replica. A failover partner cannot host a secondary replica for the same availability group.
- Failover cluster instances support only manual failover. You cannot configure AlwaysOn automatic failover to a replica on a failover cluster instance.
- Failover cluster instances do not support initial data synchronization by using the New Availability Group Wizard, Add Database To Availability Group Wizard, or Add Replica To Availability Group Wizard.

If you are using a failover cluster instance with AlwaysOn Availability Groups, you must prepare the secondary database on the instance by using a different method, such as backup and restore, and then join that secondary database to the availability group.

> ***MORE INFO*** **FAILOVER CLUSTER INSTANCES AND ALWAYSON**
>
> **You can learn more about failover cluster instances and AlwaysOn at *http://msdn.microsoft.com/en-us/library/ff929171.aspx*.**

PRACTICE Deploying AlwaysOn Availability Groups

In this practice, you deploy AlwaysOn Availability Groups.

EXERCISE 1 Prepare for AlwaysOn Availability Groups

In this exercise, you prepare the servers for the deployment of AlwaysOn Availability Groups. To complete this exercise, perform the following steps:

1. On the domain controller, edit the SQL-POLICY GPO.
   - **A.** Add an Isolation connection security rule that requires authentication for inbound connections and requests authentication for outbound connections by using Computer (Kerberos V5) authentication for all profiles.
   - **B.** Create an Inbound Port–based rule that allows TCP traffic on all local ports if the connection is secure and comes from computers DC, SQL-A, SQL-B, SQL-C, SQL-D, and SQL-CORE, as shown in Figure 8-24. Enable this rule in all profiles.

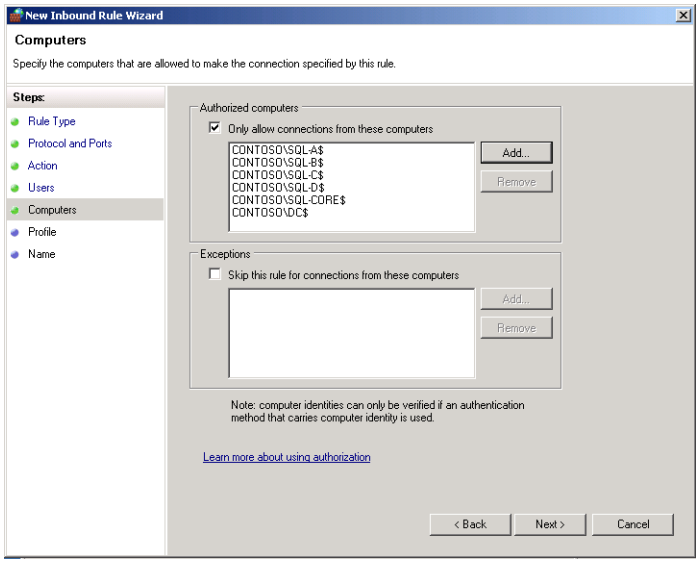

FIGURE 8-24 Isolation rule

C. Create an Inbound Port–based rule that allows UDP traffic on all local ports if the connection is secure and comes from computers DC, SQL-A, SQL-B, SQL-C, SQL-D, and SQL-CORE. Enable this rule in all profiles.

2. Log on to servers SQL-C and SQL-D with the Kim_Akers user account.
3. Run **gpupdate /force** from an elevated command prompt to apply the new firewall rule to these computers.
4. Create a new shared folder named **Share** on SQL-C in the C:\Share directory. Configure the share so that the Contoso\SQL-CLUSTER user has read\write access.
5. Install a new Database Engine instance named **AlwaysOn** on SQL-C and SQL-D. Configure Contoso\SQL-Cluster as the SQL Server service account and configure Kim_Akers as the SQL Administrator on these instances by running the following command from an elevated command prompt on each server:

```
setup.exe /qs /ACTION=Install /FEATURES=SQLEngine /INSTANCENAME=AlwaysOn /
SQLSVCACCOUNT="CONTOSO\SQL-CLUSTER" /SQLSVCPASSWORD="Pa$$w0rd" /
SQLSYSADMINACCOUNTS="Contoso\Kim_Akers" /AGTSVCACCOUNT="CONTOSO\SQL-CLUSTER" /
AGTSVCPASSWORD="Pa$$w0rd" /IACCEPTSQLSERVERLICENSETERMS
```

6. On server SQL-C, use SQL Server Management Studio to connect to the SQL-C \AlwaysOn instance. Right-click the SQL-C\ALWAYSON node and click Start PowerShell. Create a mirroring endpoint on this instance by using the following commands:

```
$endpoint = New-SqlHadrEndpoint AlwaysOnEndpoint -Port 7026 -Path
   SQLSERVER:\SQL\SQL-C\ALWAYSON
Set-SqlHadrEndpoint -InputObject $endpoint -State "Started"
```

7. On server SQL-D, use SQL Server Management Studio to connect to the SQL-D \AlwaysOn instance. Right-click the SQL-C\ALWAYSON node and select Start PowerShell. Create a mirroring endpoint on this instance by using the following commands:

```
$endpoint = New-SqlHadrEndpoint AlwaysOnEndpoint -Port 7026 -Path
   SQLSERVER:\SQL\SQL-D\ALWAYSON
Set-SqlHadrEndpoint -InputObject $endpoint -State "Started"
```

8. Use SQL Server Configuration Manager to enable AlwaysOn Availability Groups on the SQL Server (ALWAYSON) service on SQL-C and SQL-D.
9. Use SQL Server Configuration Manager to restart the SQL Server (ALWAYSON) service on SQL-C and SQL-D.

EXERCISE 2 Create a Database and Add It to a New Availability Group

In this exercise, you create a database and add it to a newly created availability group. To complete this exercise, perform the following steps:

1. On the SQL-C\ALWAYSON instance, create a new database named **Jupiter** by using the default settings.
2. Perform a full backup of database Jupiter.
3. Use the New Availability Group Wizard to create a new availability group named **Availability Group Alpha**.
4. Add the Jupiter database to the new availability group.
5. Ensure that SQL-C\ALWAYSON and SQL-D\ALWAYSON are configured as replicas.
6. Choose Full as the data synchronization preference and use the \\SQL-C\Share share as the accessible network location.

EXERCISE 3 Create an Availability Group Listener

In this exercise, you create an availability group listener for the availability group you configured in the previous exercise. To complete this exercise, perform the following steps:

1. Use SQL Server Management Studio to create an availability group listener for the Availability Group Alpha availability group that uses the following properties:
   - Listener DNS Name: Alpha-Listener
   - Port: 7028
   - Static IP: 10.0.0.222

EXERCISE 4 Configure Availability and Failover Mode

In this exercise, you configure availability modes and failover modes and then perform manual failover. To complete this exercise, perform the following steps:

1. Configure Availability Group Alpha so that:
   - Both instances use the synchronous-commit availability mode.

- SQL-C\AlwaysOn uses the automatic failover mode.
- SQL-D\AlwaysOn is a Readable Secondary.

2. Perform manual failover from the primary to the replica instance.

Lesson Summary

- The AlwaysOn Availability Groups feature is an alternative to database mirroring.
- Availability groups are supported in production on SQL Server 2012 Enterprise edition only.
- An AlwaysOn availability group can have one primary and four secondary replicas.
- You must create mirroring endpoints either before or during the availability group creation process.
- An availability group replica can contain multiple databases.
- You can configure secondary replicas to be available to read-only queries.
- Failover occurs on a per-replica basis.
- Synchronous-commit mode involves higher transaction latency but allows manual and automatic failover.
- Asynchronous-commit mode minimizes transaction latency, is suitable for geographically dispersed clusters, but only supports forced failover.
- You can perform availability group failover by using SQL Server Management Studio, the ALTER AVAILABILITY GROUP Transact-SQL statement, or the Switch-SqlAvailabilityGroup PowerShell cmdlet.
- You can have only one listener per availability group.

Lesson Review

Answer the following questions to test your knowledge of the information in this lesson. You can find the answers to these questions and explanations of why each answer choice is correct or incorrect in the "Answers" section at the end of this chapter.

1. Which tool can you use to enable AlwaysOn Availability Groups on a SQL Server 2012 instance?
   - **A.** SQL Server Management Studio
   - **B.** SQL Server Installation Center
   - **C.** SQL Server Configuration Manager
   - **D.** SQL Server Data Tools
2. Which Windows PowerShell cmdlet can you use to perform manual availability group failover?
   - **A.** Switch-SqlAvailabilityGroup
   - **B.** New-SqlHadrEndpoint

C. New-SqlAvailabilityGroupListener

D. Enable-SqlAlwaysOn

3. Which Windows PowerShell cmdlet can you use to create a mirroring endpoint on an instance when preparing it for the deployment of AlwaysOn Availability Groups?

   A. New-SqlAvailabilityGroupListener

   B. Switch-SqlAvailabilityGroup

   C. Enable-SqlAlwaysOn

   D. New-SqlHadrEndpoint

4. You will configure an AlwaysOn Availability Group to support automatic failover from the primary replica to any available secondary replica. Which of the following availability modes should you configure for the replicas in this availability group? (Each correct answer forms part of a complete solution. Choose all that apply.)

   A. Configure the primary replica to use the asynchronous-commit availability mode.

   B. Configure the primary replica to use the synchronous-commit availability mode.

   C. Configure the secondary replica to use the asynchronous-commit availability mode.

   D. Configure the secondary replica to use the synchronous-commit availability mode.

Case Scenarios

In the following case scenarios, you apply what you have learned about SQL Server clustered instances and AlwaysOn. You can find answers to these questions in the "Answers" section at the end of this chapter.

Case Scenario 1: Failover Cluster Instances at Contoso

You are designing a failover cluster instance solution at Contoso. You will deploy a four-node failover cluster at the Melbourne site. You will also deploy a failover cluster instance that has nodes in the cities of Brisbane, Sydney, Adelaide, and Perth. Each of these cities resides on a different TCP/IP subnet.

With these facts in mind, answer the following questions:

1. Which edition of SQL Server 2012 should you deploy to support the proposed cluster configuration at the Melbourne site?
2. Which host operating systems can you use to support the proposed cluster configuration at the Brisbane, Sydney, Adelaide, and Perth sites?
3. Which tools can you use to perform manual failover when performing maintenance?

Case Scenario 2: AlwaysOn Availability Groups at Fabrikam

You are planning an AlwaysOn Availability Groups deployment at Fabrikam. Fabrikam wants to deploy replica instances in the cities of Sydney, Brisbane, Canberra, and Melbourne. Fabrikam security policy dictates that you must use local virtual accounts rather than domain security accounts for the SQL Server service accounts for each of the replica instances. The Chief Information Officer (CIO) at Fabrikam wants you to configure the AlwaysOn Availability Group so that automatic failover is possible.

With these facts in mind, answer the following questions:

1. Which editions of Windows Server 2008 R2 could you use to support the proposed configuration?
2. What factors influence the choice of authentication method for the mirroring endpoints?
3. Which availability mode should you configure on the primary and secondary replicas, given the project requirements?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following tasks.

Implement a SQL Server Clustered Instance

Prior to completing each task in the following practices, list the steps you would take to accomplish the task. After completing the task, assess how accurately you predicted the necessary steps.

- **Practice 1** Create a new database on the failover clustering instance you created during the exercises at the end of Lesson 1.
- **Practice 2** Shut down server SQL-C. Verify that the database you created on the cluster is still available on server SQL-D.

Implement AlwaysOn

Prior to completing each task in the following practices, list the steps you would take to accomplish the task. After completing the task, assess how accurately you predicted the necessary steps.

- **Practice 1** Create a new database and add it to availability group Alpha.
- **Practice 2** Delete the existing listener and create a new listener for availability group Alpha by using Transact-SQL.

Answers

This section contains the answers to the lesson review questions and solutions to the case scenarios in this chapter.

Lesson 1

1. **Correct Answer: C**
   - **A. Incorrect:** Windows 7 Enterprise edition cannot be used as the host operating system for a multi-site failover cluster instance.
   - **B. Incorrect:** Windows Server 2008 Enterprise edition does support failover clustering but does not support multi-site failover clusters.
   - **C. Correct:** Windows Server 2008 R2 Enterprise edition supports multi-site failover clustering.
   - **D. Incorrect:** Windows Vista Ultimate edition cannot be used as the host operating system for a multi-site failover cluster instance.
2. **Correct Answer: A**
   - **A. Correct:** The Move-ClusterGroup cmdlet enables you to move a clustered service or application from one node to another in a failover cluster. You can use this cmdlet to perform manual failover of a SQL Server clustered instance from one node to another.
   - **B. Incorrect:** The Move-ClusterResource cmdlet enables you to move a clustered resource from one clustered application to another but not to move a clustered service or application from one node to another.
   - **C. Incorrect:** The Move-ClusteredSharedVolume cmdlet enables you to move the ownership of a clustered shared volume from one node to another. You cannot use this cmdlet to perform failover on a SQL Server clustered instance.
   - **D. Incorrect:** The Move-ClusterVirtualMachineRole cmdlet enables you to move a clustered virtual machine to a different cluster node. You cannot use this cmdlet to perform failover on a SQL Server clustered instance.
3. **Correct Answer: A**
   - **A. Correct:** You should evict the failed node. After this is done, you can repair the server, join it back to the cluster, and then reinstall SQL Server.
   - **B. Incorrect:** You should not evict the new primary node; you should instead evict the failed node from the cluster.
   - **C. Incorrect:** You should not reinstall SQL Server on the failed node until you have evicted the node, repaired the failure, and joined the node back to the cluster.
   - **D. Incorrect:** You should not join the failed node back to the cluster until you have evicted and repaired the node.

4. **Correct Answers: A, B, and D**
   - **A. Correct:** You must run advanced cluster preparation on all nodes that will participate in the failover cluster instance.
   - **B. Correct:** You must run advanced cluster preparation on all nodes that will participate in the failover cluster instance.
   - **C. Incorrect:** You run advanced cluster completion only on the node that has control of the shared disk resource. Because SQL-B has control of this resource, you should not run this process on SYD-A.
   - **D. Correct:** You run advanced cluster completion only on the node that has control of the shared disk resource.

Lesson 2

1. **Correct Answer: C**
   - **A. Incorrect:** You can enable AlwaysOn Availability Groups by using either SQL Server Configuration Manager or PowerShell. You cannot perform this task by using SQL Server Management Studio.
   - **B. Incorrect:** You can enable AlwaysOn Availability Groups by using either SQL Server Configuration Manager or PowerShell. You cannot perform this task by using SQL Server Installation Center.
   - **C. Correct:** You can enable AlwaysOn Availability Groups by using either SQL Server Configuration Manager or PowerShell.
   - **D. Incorrect:** You can enable AlwaysOn Availability Groups by using either SQL Server Configuration Manager or PowerShell. You cannot perform this task by using SQL Server Data Tools.
2. **Correct Answer: A**
   - **A. Correct:** You use the Switch-SqlAvailabilityGroup cmdlet to perform manual availability group failover.
   - **B. Incorrect:** You use the New-SqlHadrEndpoint cmdlet to create a mirroring endpoint for AlwaysOn Availability Groups. You use the Switch-SqlAvailabilityGroup cmdlet to perform manual availability group failover.
   - **C. Incorrect:** You use the New-SqlAvailabilityGroupListener cmdlet to create a new availability group listener. You use the Switch-SqlAvailabilityGroup cmdlet to perform manual availability group failover.
   - **D. Incorrect:** You use the Enable-SqlAlwaysOn cmdlet to enable AlwaysOn on an instance. You use the Switch-SqlAvailabilityGroup cmdlet to perform manual availability group failover.

3. **Correct Answer: D**
   A. **Incorrect:** You use the New-SqlAvailabilityGroupListener cmdlet to create a new availability group listener.
   B. **Incorrect:** You use the Switch-SqlAvailabilityGroup cmdlet to perform manual availability group failover.
   C. **Incorrect:** You use the Enable-SqlAlwaysOn cmdlet to enable AlwaysOn on an instance.
   D. **Correct:** You use the New-SqlHadrEndpoint cmdlet to create a mirroring endpoint for AlwaysOn Availability Groups.
4. **Correct Answers: B and D**
   A. **Incorrect:** To support automatic failover to any available secondary replica, all replicas must use synchronous-commit mode. Automatic failover cannot occur if the primary replica uses asynchronous-commit mode.
   B. **Correct:** To support automatic failover to any available secondary replica, all replicas must use synchronous-commit mode. Automatic failover cannot occur if the primary replica uses asynchronous-commit mode.
   C. **Incorrect:** To support automatic failover to any available secondary replica, all replicas must use synchronous-commit mode. Automatic failover cannot occur if the primary replica uses asynchronous-commit mode.
   D. **Correct:** To support automatic failover to any available secondary replica, all replicas must use synchronous-commit mode. Automatic failover cannot occur if the primary replica uses asynchronous-commit mode.

Case Scenario 1

1. You must deploy SQL Server 2012 Enterprise edition to support the proposed cluster configuration because this is the only edition that supports four nodes.
2. You must support multi-subnet failover clustering, which requires either Windows Server 2008 R2 Enterprise or Datacenter editions. Windows Server 2008 does not support multi-subnet failover clustering.
3. You can use either the Failover Cluster manager or the Move-ClusterGroup PowerShell cmdlet to perform manual failover.

Case Scenario 2

1. You can use the Enterprise or Datacenter editions of Windows Server 2008 R2 to support the proposed configuration.

2. Because the SQL Server service uses local accounts, you must use certificate-based authentication for the endpoints.
3. You must configure the synchronous-commit availability mode, given the requirement for automatic failover.

Index

Numbers and Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

About the Authors

ORIN THOMAS, MCITP, MCT, MVP, is an author, trainer, and regular public speaker who has authored more than a dozen books for Microsoft Press. In addition to holding MCITP Server Administrator and Enterprise Administrator certifications, Orin is a Microsoft vTSP, and the convener of the Melbourne Security and Infrastructure Group. His most recent books are on Windows 7 and Exchange Server 2010. You can follow Orin on Twitter @orinthomas.

PETER WARD is the Chief Technical Architect of WARDY IT Solutions, a company he founded in 2005. WARDY IT Solutions has been awarded the Microsoft Data Platform Partner of the Year each year since 2009 and has been a member of the Deloitte Technology Fast 500 since 2012. Peter is a highly regarded speaker at SQL Server events, a sought-after SQL Server consultant, and a trainer who provides solutions for some of the largest SQL Server sites in Australia. He has been recognized as a Microsoft Most Valuable Professional since 2006 for his technical excellence and commitment to the SQL Server community.

BOB TAYLOR is the Principal Program Manager for the MCA/MCM program at Microsoft. He holds many industry certifications including Microsoft Certified Architect (2005/2008), Microsoft Certified Master (2005/2008), MCITP, MCSD.NET, MCT, and MCSE. boB's career in IT started over 39 years ago, when he programmed FORTRAN on punch cards (he has also written assembly language programs on a drum-based computer). He understands the entire development life cycle thanks to his experience as a line-of-business programmer, manager and Vice President of Software Engineering, and Director of Database Technologies. boB started working with SQL Server on version 4.2.1a in the early 1990s, and he has participated in the development of more than 10 Microsoft Certified Professional certification exams. You can find boB's blog at *http://blogs.msdn.com/boBTaylor*.

The unique spelling of boB's first name comes from the fact that he is a magician (*http://www.majikbybob.com*) and mentalist (*http://www.classicclairvoyant.com*)—the only things he has done longer than software development.

Contributor

NEIL HAMBLY is a database architect and SQL professional with more than 13 years' expertise in SQL Server, starting with Version 6.5 through to the very latest 2012 releases. Neil has held a number of database roles both with major organizations (BBC, ABN AMRO, ACCENTURE) and market-leading smaller companies. He is experienced in DBA, developer, and architect roles. Neil is now also a regular speaker at SQL Conferences, on webcasts, and for local user groups. The PASS chapter leader for UK SQL Server London, Neil often presents in the United Kingdom as well as at international events. He just loves learning and teaching SQL.