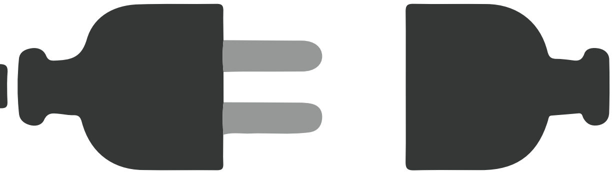
Microsoft

Microsoft System Center

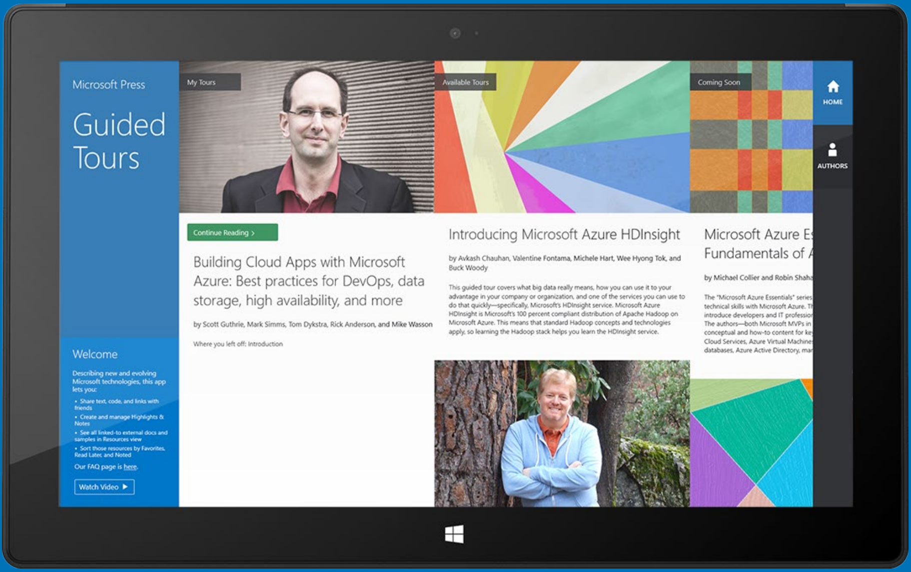
Extending Operations Manager Reporting

George Wallace ▪ Chris Jones ▪ Bill May ▪ Fred Lee
Mitch Tulloch, Series Editor



Hear about
it first.

Wait, there's more...



Microsoft Press

Guided Tours

Welcome

Describing new and evolving Microsoft technologies, this app lets you:

- Share text, code, and links with friends
- Create and manage Highlights & Notes
- See all linked-to external docs and samples in Resources view
- Sort those resources by Favorites, Read Later, and Noted

Our FAQ page is here.

Building Cloud Apps with Microsoft Azure: Best practices for DevOps, data storage, high availability, and more

by Scott Guthrie, Mark Simms, Tom Dykstra, Rick Anderson, and Mike Wasson

Where you left off: Introduction

Introducing Microsoft Azure HDInsight

by Avkash Chauhan, Valentine Fontama, Michele Hart, Wee Hyong Tok, and Buck Woody

This guided tour covers what big data really means, how you can use it to your advantage in your company or organization, and one of the services you can use to do that quickly—specifically, Microsoft's HDInsight service. Microsoft Azure HDInsight is Microsoft's 100 percent compliant distribution of Apache Hadoop on Microsoft Azure. This means that standard Hadoop concepts and technologies apply, so learning the Hadoop stack helps you learn the HDInsight service.

Microsoft Azure Es
Fundamentals of A

by Michael Collier and Robin Shaha

The "Microsoft Azure Essentials" series
technical skills with Microsoft Azure. Th
introduce developers and IT profession
The authors—both Microsoft MVPs in
conceptual and how-to content for ke
Cloud Services, Azure Virtual Machine
databases, Azure Active Directory, ma

Find more great content and resources in the **Microsoft Press Guided Tours** app.

The Microsoft Press Guided Tours app provides insightful tours by Microsoft Press authors of new and evolving Microsoft technologies.

- Share text, code, illustrations, videos, and links with peers and friends
- Create and manage highlights and notes
- View resources and download code samples
- Tag resources as favorites or to read later
- Watch explanatory videos
- Copy complete code listings and scripts

Download from
Windows Store

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

Library of Congress Control Number: 2014955192
ISBN: 978-0-7356-9578-8

Printed and bound in the United States of America.

First 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://aka.ms/tellpress.

This book is provided "as-is" and expresses the author's views and opinions. The views, opinions and information expressed in this book, including URL and other Internet Web site references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

Acquisitions Editor: Karen Szall
Developmental Editor: Karen Szall
Copyeditor: Megan Smith-Creed
Cover Illustration: Twist Creative, Seattle

Contents

This page intentionally left blank

Introduction

The System Center Operations Manager data warehouse stores many objects, making it possible to report on performance, availability, configuration, and security. You can use data collected by Operations Manager for issue tracking, awareness, planning, and forecasting, all of which are important factors for maintaining and managing your environment. But understanding what to report on and how to do so is what actually contributes to the stability of your infrastructure. If you're not using the data you collect about your environment, you're simply wasting space. If you use Operations Manager to monitor your environment, whether your infrastructure or a public, private, or hybrid cloud, this book is a good resource to help you understand the basics of reporting and how to build queries and stored procedures for your reports. It can also help you understand the various elements of Operations Manager management packs that are used when you create reports.

This book assumes that you have a good understanding of Operations Manager and the management pack structure required to create custom reports. It also assumes that you have a good working knowledge of SQL Server and tools like Report Builder, Business Intelligence Development Suite, and SQL Server Data Tools.

About the companion content

The companion content for this book can be downloaded from the following page:

http://aka.ms/SCopsmgr/files

The companion content includes the following:

- Sample SELECT query (Chapter 5)
- Contoso example report library and report fragment (Chapter 7)
- T-SQL code listings (Chapter 6)
- Microsoft Excel Power View dashboard example (Chapter 8)

Acknowledgments

The authors would like to specifically thank Kevin Holman and Alban Montanera for their contributions to this book, and also the great Operations Manager community within Microsoft and around the globe. There truly is a great passion when it comes to Operations Manager.

The Series Editor would like to thank the following individuals at Microsoft who reviewed the outlines for the proposed titles in this series and provided helpful feedback to the authors: David Ziembicki, Adam Fazio, Robert Larson, David Stoker, and Joel Yoker.

Free ebooks from Microsoft Press

From technical overviews to in-depth information on special topics, the free ebooks from Microsoft Press cover a wide range of topics. These ebooks are available in PDF, EPUB, and Mobi for Kindle formats, ready for you to download at:

http://aka.ms/mspressfree

Check back often to see what is new!

Errata, updates, & book support

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

http://aka.ms/SCopsmgr/errata

If you discover an error that is not already listed, please submit it to us at 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 and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *http://support.microsoft.com*.

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://aka.ms/tellpress

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*.

CHAPTER 1

Operations Manager and the cloud

To effectively optimize your IT environment, you need to understand how it is performing at all layers of the Open Systems Interconnection (OSI) model. Whether you're managing on-premises datacenters or leveraging cloud-based solutions, being able to analyze and report on your underlying infrastructure, applications, and business services is a critical factor of a well-managed IT environment. You need to have a method for identifying potential outages and service interruptions, especially if you are providing a service or a number of services to customers. Microsoft System Center is a key part of the Microsoft cloud vision, and System Center Operations Manager provides you with a deep look into your IT environment. Although the main focus of this book is Operations Manager Reporting, it's important that you begin by understanding the larger picture of how Operations Manager and other types of Microsoft reporting solutions fit into the overall Microsoft strategy for the cloud.

Microsoft cloud solutions

In recent years, the focus of IT has evolved toward big data, the cloud, and mobile computing. This transformation has led to businesses being required to manage and deliver highly scalable IT services, which has significantly impacted how they operate. The challenges of supporting new applications, the proliferation of devices, and the abundance of data requires businesses to find cost-effective solutions for managing their IT environments. Microsoft cloud solutions provide the ability to consume, manage, and visualize computing resources based on the following models:

- **Datacenter** A datacenter is an IT environment operated and managed either on-premises or through an enterprise service provider. Datacenters are used to distribute large amounts of data between networked sites and the customer base. Many businesses and government agencies have leveraged cost-efficient datacenters to secure and manage their data and support continuity of operations planning initiatives.

- **Microsoft Azure** Microsoft Azure is the Microsoft cloud computing platform and infrastructure. Microsoft Azure provides solutions to build, deploy, and manage applications and services through global datacenters managed by Microsoft. Microsoft Azure can be summarized as the abstraction of hardware and software resources within and between datacenters.
- **Hybrid cloud** A hybrid cloud combines on-premises IT resources (private cloud) with Microsoft cloud solutions (Microsoft Azure) to enable a seamless experience of resources in the public and private cloud for the end user. Businesses can either fully migrate their private cloud or choose to transition business-critical or consumer-facing software and services to Microsoft Azure. The hybrid cloud solution is a favorite approach for businesses that want cost-efficiency and resiliency for business operations in their cloud-computing solution.

Each of these models enables a cloud-first approach to a centralized management of IT resources, independent of the physical locations of the business services or data. Businesses can provide on-demand access to IT resources based on real-time or forecast consumption. Cloud computing transforms the datacenter into a highly customizable model based on delivering agile workload provisioning and scalability and optimizing the use and consumption of IT resources, all while providing a secure and continuous IT environment.

Integration with Microsoft cloud solutions

With Microsoft cloud solutions focused on seamlessly integrating the physical, virtual, and cloud infrastructures hosted within private, Microsoft Azure, or hybrid clouds, Operations Manager plays a critical role in measuring performance, validating service and operational levels, and planning capacity. Operations Manager Reporting can be used to make sense of all of the various application, performance, and operational data collected via Operations Manager, specifically the Microsoft Monitoring agents. All of these solutions are tightly integrated and allow you to develop data-rich reports using the tools discussed within this book. This will enable you to create useful visual representations of your IT enterprise whether it is hosted locally, spans Microsoft cloud solutions, or is a combination of the two.

Before moving on to how Operations Manager reports are created and delivered, it might be helpful to review a couple of the product offerings specifically developed to integrate with Microsoft cloud solutions. As discussed earlier, Operations Manager seamlessly integrates with the entire System Center suite to transform the concept of a traditional datacenter and to change the understanding of how and where an application is hosted and consumed. For example, integrating Operations Manager with additional System Center programs, such as Service Manager, Virtual Machine Manager, and System Center Orchestrator, enables businesses to visualize their IT service metrics against operational data through reports delivered through on-demand mechanisms or presented in websites, portals, or even dashboards. In addition, the integration of Operations Manager and Visual Studio enables IT administrators to deliver reports directly to developers to enhance communication and visibility between operations and development teams. The following solutions detail how Operations Manager Reporting integrates with Microsoft cloud solutions:

- **Service Reporting** Service Reporting is an optional feature installed with System Center Orchestrator. Used alongside the Windows Azure Pack for Windows Server, Service Reporting provides usage data about virtual machines at both the IT administrator and customer or tenant level. Service Reporting enables you to create detailed views displaying the consumption of computational, memory, storage, and networking resources for virtual machines provisioned through the Microsoft cloud solutions. Service Reporting uses aggregate data collection from both Operations Manager and the Windows Azure Pack to further extend the reporting capabilities of Microsoft cloud solutions. For more information on Service Reporting in System Center 2012 R2, see *http://technet.microsoft.com/en-us/library/dn251058.aspx*.
- **Global Service Monitor** Global Service Monitor (GSM) is a feature of Operations Manager that provides a method to monitor web-based applications from multiple locations around the world. GSM is incorporated within the Operations Manager console to allow you to monitor both internal-facing and external-facing web applications from a consolidated user interface. Both IT administrators and developers can use the Operations Manager console to leverage GSM from the end-user perspective. As a cloud-based service, GSM enables you to monitor application failures or performance issues from the customer perspective based solely on a geographic location. For more information on Global Service Monitor, see *http://technet.microsoft.com/en-us/library/jj860368.aspx*.

These are just two examples of expanding how Operations Manager integrates with Microsoft cloud solutions. Using Operations Manager Reporting, you can use not only the native SQL Server Reporting Services tools, you can also leverage the additional tools and applications presented within this book to create powerful, data-driven reports. How this data is collected, consumed, and presented is determined by the audience. For instance, creating and delivering reports to IT operations teams or executive-level leadership, your audience will determine how well your reports will be received.

The following are some examples of how the integration of Operations Manager Reporting with Microsoft cloud solutions can positively impact your IT enterprise:

- **IT service reporting** Both Operations Manager and Service Manager provide data warehouses to enable customers to develop and publish data about the IT environment and services, for example, service-level metrics detailing application performance, uptime metrics, and performance analysis for an application owner. This data can be presented to the customer in many different ways using familiar tools or published directly to the intranet or an internet facing website.

- **Business intelligence** Through Operations Manager Reporting, IT administrators can deliver subscription-based reports to application owners and empower these users to enhance their analysis of operational data using familiar business intelligence tools. User-centric data can be analyzed and shared throughout the IT environment either through business intelligence features of Microsoft Excel or through more advanced analytics using SQL Server. By providing powerful business intelligence features in a unified, familiar set of tools, customers can develop creative reporting solutions without a steep learning curve. By leveraging other System Center products like System Center Service Manager, you can leverage the analytic cubes provided by Service Manager to further enhance the data you can represent from Operations Manager collected data.

Operations Manager

It is important for you to first understand how Operations Manager fits within Microsoft System Center before you start developing reports. System Center is a collection of products that deliver unified management across Microsoft cloud solutions deployed in on-premises datacenters, service-provider-managed datacenters, in Microsoft Azure environments, or across hybrid clouds. The following System Center aspects enable the traditional datacenter to embrace the Microsoft cloud solution:

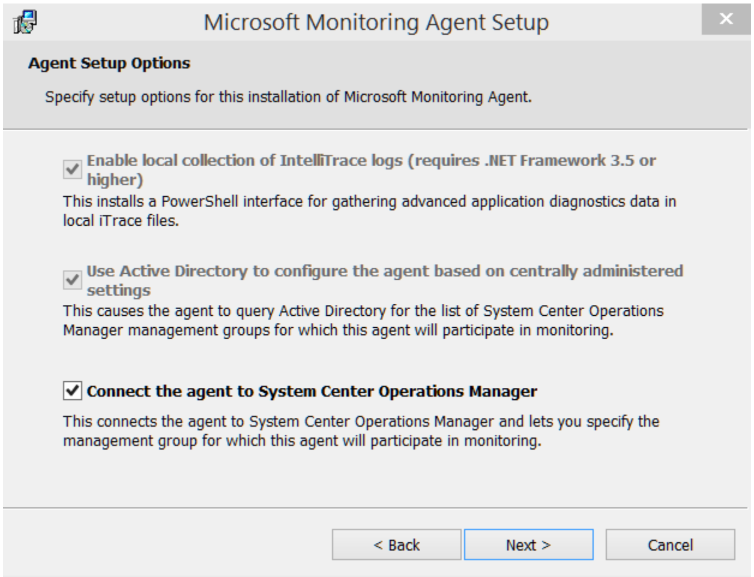
- **Best in class** Developed with in-depth knowledge of each Microsoft enterprise-level product, such as Windows Server, Exchange Server, and SQL Server, System Center delivers the best methods and practices for managing Microsoft products and services.

- **Simple and cost-effective** System Center reduces the complexity of datacenter operations by providing a unified solution to deploy and manage infrastructure, application, and service resources.

- **Cloud focused** System Center optimizes cloud computing workloads throughout their lifecycle. Whether you are managing a private cloud, a Microsoft Azure environment, or a hybrid cloud, with the System Center cloud-first strategy you can provision, scale, and maintain the resiliency of IT resources across clouds.

System Center provides a consistent experience for managing Microsoft clouds through application management, service delivery, automation, and infrastructure management. These core capabilities enable easy access to IT services and allow application owners and IT administrators to realize the benefits of cloud computing whether they deploy private, Microsoft Azure, or hybrid cloud environments. System Center applications are tightly integrated with one another and provide IT decision-makers, application managers, and end-users or customers with a rich, robust reporting solution. Operations Manager plays a key role within this strategy by providing application and infrastructure monitoring capabilities paired with a robust reporting solution which allows you to visualize operational, performance, and capacity metrics within the IT enterprise.

The Microsoft Monitoring Agent 2013, introduced in System Center 2012 Operations Manager R2, includes the IntelliTrace Collector, which is a feature originally introduced with Microsoft Visual Studio. The Operations Manager .NET Application Performance Monitoring (APM) capabilities have been expanded to include the full functionality of the IntelliTrace Collector. During a manual installation of the Microsoft Monitoring Agent 2013, you may notice on the Agent Setup Options page, there is an option to enable local collection of IntelliTrace logs, as shown in Figure 1-1. This option is enabled by default and you cannot deselect this option when you use the manual installation method.


Microsoft Monitoring Agent Setup

Agent Setup Options

Specify setup options for this installation of Microsoft Monitoring Agent.

☑ Enable local collection of IntelliTrace logs (requires .NET Framework 3.5 or higher)

This installs a PowerShell interface for gathering advanced application diagnostics data in local iTrace files.

☑ Use Active Directory to configure the agent based on centrally administered settings

This causes the agent to query Active Directory for the list of System Center Operations Manager management groups for which this agent will participate in monitoring.

☑ Connect the agent to System Center Operations Manager

This connects the agent to System Center Operations Manager and lets you specify the management group for which this agent will participate in monitoring.

FIGURE 1-1 The Agent Setup Options dialog box of the Microsoft Monitoring Agent

The inclusion of the IntelliTrace feature means that Operations Manager is no longer limited to the realm the IT systems and network administrators because development teams can now use Operations Manager to monitor applications written using the .NET framework. Furthermore, Operations Manager provides the tools needed to analyze these collected application and performance metrics and to present them in various visual formats using Operations Manager Reporting.

NOTE **The Microsoft Monitoring Agent 2013 can be downloaded from http://www.microsoft.com/en-us/download/details.aspx?id=40316.**

Operations Manager Reporting

Operations Manager Reporting collects and aggregates operational data into a dedicated data warehouse for historical reporting purposes. The Operations Manager data warehouse is the data repository used by Operations Manager Reporting to deliver reports on demand or on a scheduled basis. Not only does this help IT administrators and application owners review and forecast performance and plan for capacity, it also allows you to create custom reports that can be developed to integrate seamlessly into the existing environment or to populate IT operational dashboards.

Reporting has been one of the most underutilized features of Operations Manager, not due to limitations, but rather because the feature requires familiarity with Microsoft SQL Server, specifically Microsoft SQL Server Reporting Services. Microsoft SQL Server provides data and presentation layers to Operations Manager Reporting. Although Operations Manager provides several standard reports, developing custom reports or leveraging Operations Manager reports has typically been limited to IT administrators and developers with an intermediate knowledge of and proficiency with Microsoft SQL Server. Without this core knowledge, developing custom reporting solutions within Operations Manager Reporting can be a challenge. Throughout this book, we will help you gain a better understanding for creating your own custom Operations Manager reports.

CHAPTER 2

Operations Manager reporting basics

After reading this chapter, you will be able to:

- Describe what you can report on with Operations Manager
- Know the main reporting elements
- Plan reports
- View reports
- Schedule reports

Reporting with Operations Manager

System Center 2012 Operations Manager is a highly configurable and extensible application that provides administrators and service providers with a holistic monitoring solution. You can use Operations Manager to report on the following:

- **Alerts** Alerts that have been triggered from rules and or monitors
- **Availability** Details of the availability or uptime of a monitored object
- **Configuration** Changes to the properties of a monitored object
- **Events** Events created on monitored objects, such as Windows event logs
- **Performance** Analysis of system and application performance of a monitored object
- **Security** Access, account information, policy, integrity, and usage information (requires Operations Manager Audit Collection Services) element

NOTE **The ability to report on the items above requires that each system and/or network device is monitored by Operations Manager. The agent must be in a monitored state and the appropriate management packs must be installed and configured in accordance with the vendors' management pack guidance.**

The built-in reporting capabilities of Operations Manager allows you to identify various performance and health trends along with configuration changes throughout your environment. It can even be used to track compliance requirements, such as service level agreements (SLA), data access, and security violations, when used in conjunction with the Audit Collection Services feature of Operations Manager.

Scenario

Reporting allows you to identify areas or systems in your environment that require attention or improvements to prevent outages or service interruptions to your end users and customers. Here is a hypothetical scenario to illustrate this:

Robert Hernady is an administrator for Litware, Inc. He recently implemented a new cloud file storage system, which is used for providing external clients with a method of storing and transferring secure data files from one customer to another. As with virtually any file storage system, free space is critical to its availability. Over the past few weeks, Robert has noticed the cloud storage share has been running out of free space. This requires him, daily, to log in to the server and delete old files, ensuring there is enough space so his customers are not impacted.

With the weekend approaching and concerned about a potential impact to his service level agreements (SLA), Robert needs to quickly mitigate the situation. Leveraging the basic, built-in reporting capabilities of Operations Manager, he can easily provide his storage engineers and change control team with valid data to support his request for additional storage.

Robert's ability to provide a historical trend of free disk space is a very basic example of how the Operations Manager reporting capabilities can be leveraged, but it shows how reporting can help you avoid what could be costly outage or service interruptions to your customers. Using the built-in features in Operations Manager, you can proactively monitor and report events such as this and send email notifications to key personnel when thresholds are reached or breached. The reporting capabilities included in Operations Manager can help avoid outages or interruptions with a proactive approach rather than a reactive one. Think of it as an advanced warning system for your infrastructure. Chapter 3, "Working with reports," will cover the benefits of creating reports in further detail.

Reporting elements

The following list describes the elements that make up Operations Manager Reporting and how they tie into the reporting structure.

- **OperationsManagerDW database** The data warehouse, simplistically speaking, is the long-term storage database for Operations Manager. This data is leveraged to produce historical and trending reports.
- **OperationsManager database** This database stores data for a short period of time, typically only a few hours or days, before it is transferred to the data warehouse for archiving purposes. This is the data that you see when looking at views in the Monitoring workspace of the console.

NOTE **The database names above are the default names used during the Operations Manager 2012 R2 installation and may vary in your environment.**

- **Report Server** The integration element between the Operations Manager databases and the Operations Manager console. Without this feature installed, you will not see Reporting in the console Navigation area.
- **Microsoft SQL Server Reporting Services (SSRS)** Provides users with a quick and easy way to aggregate the data stored in the OperationsManager and OperationsManagerDW databases to generate reports.
- **Report files** The collection of reports found in the Reports Library. The report files are contained in management packs and are installed when management packs are imported into Operations Manager.

NOTE **Not all management packs contain report files. Creating custom reports and bundling them with management packs is covered in several later chapters of this book.**

See also *For a complete list of reports installed with Operations Manager 2012 R2, see http://technet.microsoft.com/en-us/library/hh212929.aspx.*

Planning reports

Operations Manager administrators are often asked to create custom reports for customers. Unfortunately, there is not a single, established methodology that works for everyone. This section describes some methodologies you may find useful. Whatever methodology you use, it's always recommended to have a plan in place before you start the report creation process. Your customer will appreciate it, and it will likely make the process smoother.

Target audience

It's critical first to understand who you are creating the report for. It will make your report building experience much easier if you take some time to understand the customer's thought process and goals. For example, an executive is likely looking for high-level information. Mid-level managers, in contrast, typically want to see data specific to their areas of operation, and technical staff generally care only about the raw data.

Requirements gathering

Writing a report without any requirements is illogical; it just doesn't make sense. It's therefore very important to gather the report requirements before you begin because it will save you a lot of time. If you are unfamiliar with the requirements gathering process and not sure what to ask, the following list of questions provides a good starting point for better understanding what your customer is expecting.

- **What is the purpose of the report?** If you have a better understanding of what the customer is trying to achieve with the report, it will give you a better idea of what data you need to retrieve from the databases.
- **What data is required in the report?** This question goes hand in hand with the previous one. When you have a good understanding of how the report will be used, you and the customer can discuss what data is available and what should be included in the report.
- **Does the report need to cover a specific time span?** You should identify whether the customer needs to be able to adjust the date range or if they require a rolling time frame, such as the past 24 hours.
- **Does the report require drill-down capabilities?** Sometimes a customer wants to drill down to more specific data about a particular object. For example, a customer may want to drill down in a report about exchange servers showing mail flow problems to determine what specific server is having problems.
- **How often does the customer want or need the report?** This question helps identify the scheduling of the report. Keep in mind that if the customer wants to use a custom date range, it must be manually selected, so the report may not be automatically delivered.
- **How should the report be delivered?** Operations Manager commonly delivers reports via email or file shares. Always validate email addresses and file share permissions to ensure the customer can receive the report on time.

- **What method will be used to make revisions?** Should the customer come directly to you, use the in-house change control process, or stop by your desk and request the change? Whatever you decide, be sure to follow up with an email confirming the requests to prevent misunderstanding.

- **What languages does the report need to support?** Translating reports into other languages is referred to as localizing. This topic is covered further in Chapter 7, "Building management packs for reporting" later in this book.

- **Does the report require specific colors or types of charts?** Be sure to ask whether the customer wants to see raw data or if they want the data presented as something more visually appealing, such as graphs or charts.

- **What are the conditions of satisfaction for the report?** This is how the overall report project will be measured. Satisfactory conditions might include that the report runs without error; that the delivery method is acceptable; and that the data is valid and in line with what the customer requires. With this question, you are trying to determine what constitutes completion of the report.

Presentation

If you are writing a report for yourself, you are probably not overly concerned about how it looks as long as it provides the data you are looking for. Customers, however, generally care about how their reports look. You might want to use the old-fashioned paper and pencil method to quickly sketch out how the report should look. Following this approach can smooth the report creation process. Keep in mind things such as sizing and margins, especially if the report will be printed.

Technical jargon

When you retrieve data from the database, it is sometimes returned using the table name. For those who are not familiar with Operations Manager, such table names may not make sense. For example, if you're creating a report to show data about alerts, you're probably going to be using the AlertStringName and AlertStringDescription fields. For those familiar with Operations Manager, such terms make sense, but they probably won't for your customer. It's therefore a good practice to rename table names like these in your reports, for example to Alert Name and Alert Description. It's certainly not necessary, but it just clarifies things a bit for the report recipients and will help them make more sense of what they are looking at in the report.

Performance

Depending on your query and the performance of the Operations Manager SQL Server database, you should test your queries to ensure they return the data in an acceptable timeframe. Queries that take a long time to process could impact SQL server performance and may also delay the report generation or even cause timeouts, which may cause a delay in report delivery.

Report parameters

If the customer wants the report generated automatically, you cannot customize parameters. If the customer wants both customization and automatic delivery, consider creating at least two separate reports: one configured with customer-selected parameters and the other with some agreed-upon parameters.

Follow up

Check in with your customers along the way to show them your progress and ask for feedback. This will help ensure that you are on the right track with the design and layout of the report. Make sure to follow up with an email summarizing the agreed-upon items.

Viewing reports

You can view the available reports in Operations Manager from within the console or from a supported web browser.

Viewing reports in the console

To view reports from within the console, click Reporting from within the navigation pane. If you do not see Reporting as shown in Figure 2-1, you should validate that the SSRS and the Operations Manager Reporting Server roles are installed and configured properly. It is important to note that it may take several minutes for the Reporting feature to show up in the console and also for the reports to show up in the Report Server.

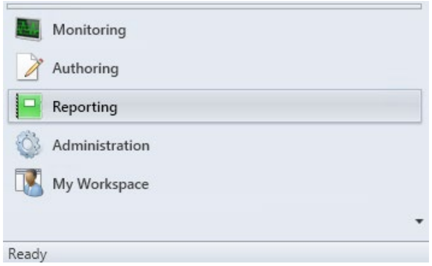
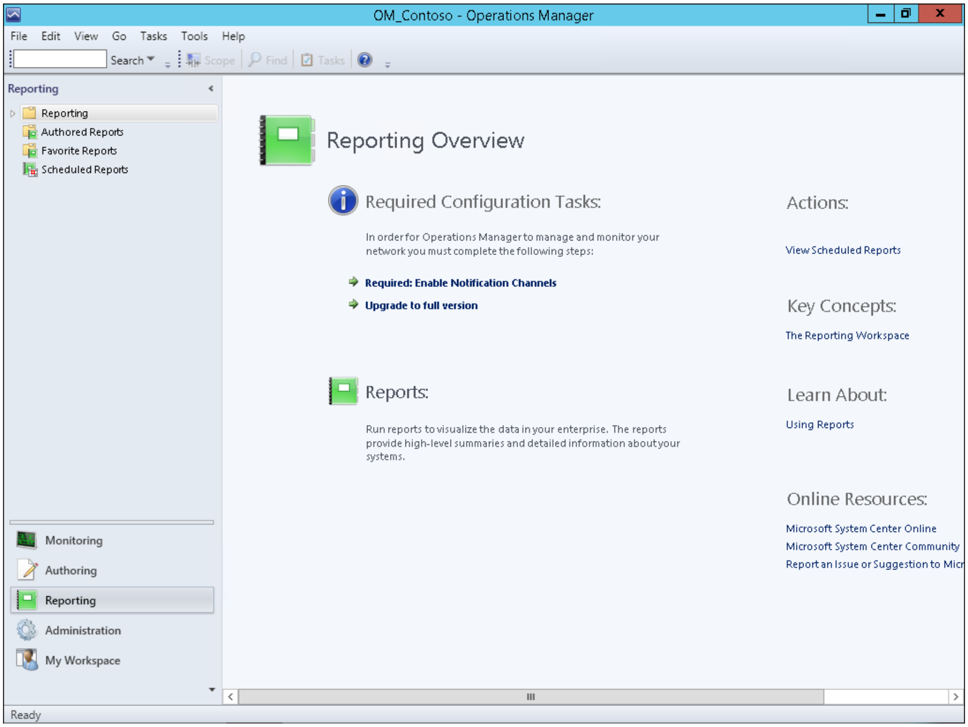


FIGURE 2-1 Operations Manager navigation pane

After you select the Reporting option, the Reporting Overview window opens, as shown in Figure 2-2. From this workspace you can select and run the various reports that are available in your Operations Manager environment. You can also configure Authored Reports and Favorite Reports and schedule reports to send on a recurring basis. Report scheduling is discussed in more detail in the next section of this chapter.



In order for Operations Manager to manage and monitor your network you must complete the following steps:

Run reports to visualize the data in your enterprise. The reports provide high-level summaries and detailed information about your systems.

FIGURE 2-2 Operations Manager Reporting workspace

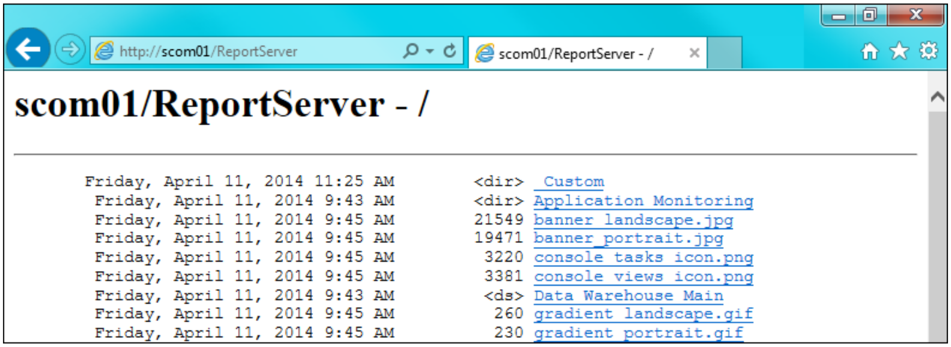
Expanding the root Reporting folder displays all of the available reports in the Reports Library. The Authored Reports folder is where you can configure reports in the Reports Library with prepopulated parameters that you can easily share with others. After running the report, click File and then Publish, give the report a custom name that identifies its purpose, and then click OK. The report will appear in the Authored Reports section.

The Favorite Reports folder allows you to save configured reports for quick and easy retrieval. To add any report to your favorites, click File and then Save To Favorites. Keep in mind that you must first run the report before the Save To Favorites option is available.

> ***See also*** *For guidance with the installation and configuration of the Operations Manager Reporting Server, go to http://technet.microsoft.com/en-us/library/hh298611.aspx.*

Viewing reports in a web browser

While it's not recommended to access report content directly from a web browser, you can certainly do so if required, for example, for troubleshooting purposes. You can validate that report files are accessible by navigating to your report server's report URL. In most cases, this will be http://<reportservername>:<portnumber>/ReportServer. For example, if your report server name is SCOM01 and you left the default port option, you would navigate to http://scom01:80/ReportServer. This would return a page similar to the one in Figure 2-3. If you are using standard ports for your report server, such as 80 or 443, the port number in the URL is typically not required.



scom01/ReportServer - /

```
Friday, April 11, 2014 11:25 AM        <dir>  Custom
 Friday, April 11, 2014 9:43 AM        <dir> Application Monitoring
 Friday, April 11, 2014 9:45 AM        21549 banner landscape.jpg
 Friday, April 11, 2014 9:45 AM        19471 banner portrait.jpg
 Friday, April 11, 2014 9:45 AM         3220 console tasks icon.png
 Friday, April 11, 2014 9:45 AM         3381 console views icon.png
 Friday, April 11, 2014 9:43 AM         <ds> Data Warehouse Main
 Friday, April 11, 2014 9:45 AM          260 gradient landscape.gif
 Friday, April 11, 2014 9:45 AM          230 gradient portrait.gif
```

FIGURE 2-3 Report Server webpage

This page presents the directory structure of your report server. Notice that the folders and report files match what you see in the operations console.

TIP In some cases, you may want to add additional reports or report libraries to your Operations Manager environment. If you choose to do this, it is very important to remember that these changes will be reflected in the Reporting workspace of the operations console and anyone with Reporting Operator rights will have access to these reports. It is recommended to bundle your custom reports into a management pack file to keep things neatly together and portable. This process is examined in further detail in Chapter 7.

If you decide to create your own reports in Operations Manager, it is recommended that you first create a custom folder to store all of your custom reports. You should never edit the default or built-in reports that are supplied within a management pack or any reports provided to you by a vendor. Instead, you should make a copy of the report and store the copy in your custom folder.

You can create a new folder or set of folders for your reports by navigating to http://<reportservername>/Reports. The returned page will look similar to the one in Figure 2-4. To create a folder or set of folders to store your custom reports, click New Folder and provide a name for the folder.

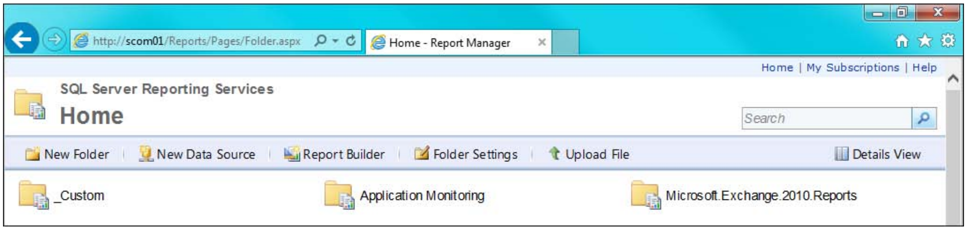


FIGURE 2-4 Report Server Reports webpage

Branding reports

The image files shown earlier in Figure 2-3 are the default graphics for reports. You can customize these by replacing them with your own graphics to brand your reports. It is important, however, to understand that if you change these image files, the new graphics will be reflected in all newly run reports. To ensure your graphics display properly, use the same dimensions as the default graphics. Also, you should not edit the default image files directly; instead, save them to a safe location in case you need to revert back to them later. To ensure a smooth transition to new graphics, name the new graphic files the same as their respective defaults.

Scheduling reports

Scheduling is useful if you want reports run periodically and sent to a user or group of users regularly. Some examples of the types of reports that customers often want to have regularly scheduled include:

- Expiring certificates (monthly)
- Servers with low disk space (daily)
- Servers rebooted overnight (daily)
- SQL Server database space (daily)

When it comes to scheduling reports, it's important to set the parameters correctly. In most cases, you can address wrong parameters passed to the report by simply editing the report. But that may not be practical if you need to configure the report quickly for upper management. A good tip is to use some advice from woodworking methodology: measure twice, cut once. This will save you from having to do it all over again. Specifically, always double check your parameters before scheduling your report.

To schedule a report, right-click the report and click Schedule to start the Delivery Settings wizard. Complete the wizard by providing a description for the report and choosing a delivery method. By default, the available delivery methods include Windows File Share, Null Delivery Provider, and Email.

TIP **If Email delivery is missing from the delivery method options; make sure the SMTP settings are properly configured in the Email Settings of your SQL Server Reporting Services Configuration Manager.**

Email is a common delivery option because it is an easy way to quickly provide a report to a number of users. If you select the Email option, you must provide the recipients' email addresses and a reply-to email address. The reply-to email address is typically an invalid or unmonitored email address, such as noreply@contoso.com or something similar to emphasize to the recipient that they shouldn't reply to the email directly.

If you want the recipients to receive the report as a PDF or Microsoft Excel file, ensure that the Include Report check box is selected and select the format type from the Render Format drop-down list. Otherwise you can simply select Include Link to provide recipients with a direct link to the report.

On the Schedule page, define how often the report should run. If your customers are expecting to receive the report by a specific time, you should schedule it a few minutes ahead of time to ensure timely delivery.

Finally, on the Parameters page, provide the context of recurring events, such as the date span or the targeted or monitored systems and devices. The Parameters area will vary by report since this page presents the parameters required to run the report. If you need to create additional or custom parameters for your reports, see *http://technet.microsoft.com/en-us/library/gg697751.aspx* for helpful guidance.

TIP **One question that is often asked is what a Null Delivery Provider is and why one would use it. The Null Delivery Provider is a feature of SSRS that allows you to schedule a report to run at a specific time and cache the contents in the ReportServerTempDB. When the customer runs a report, it runs against the cached data which allows the report to generate much more quickly. For example, imagine that you have a report that takes 4 or 5 minutes to run because of the complexity of the query behind it. If you don't want your customers to have to wait that long for the report to run, you can use the Null Delivery Provider to make the report run faster.**

CHAPTER 3

Working with reports

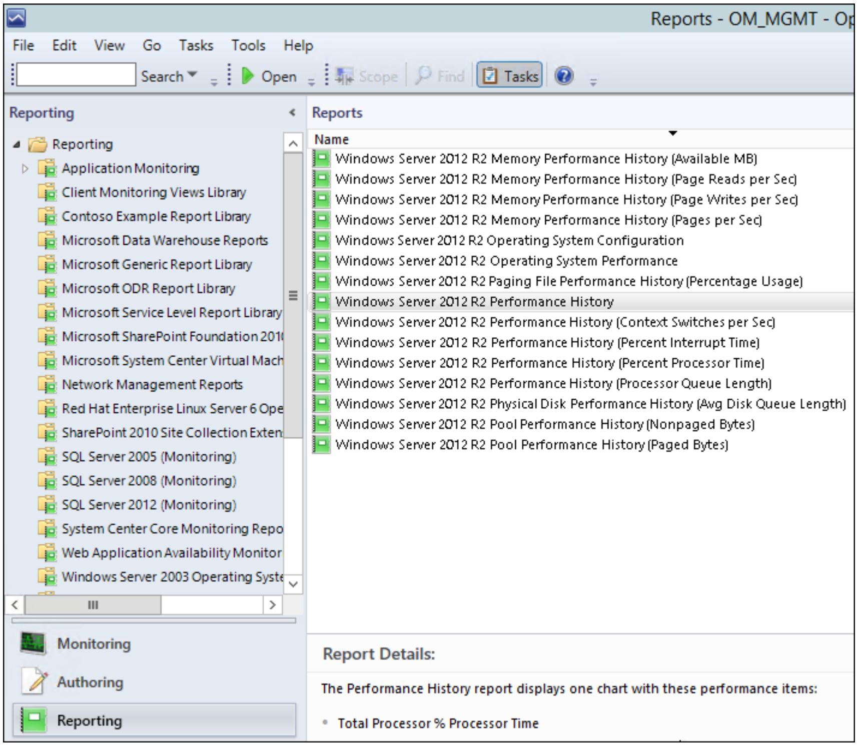
The reports in Microsoft System Center Operations Manager present the raw data collected and aggregated within the Operations Manager data warehouse into a presentation format that is not only easy to read and comprehend, but also provides a visual representation of your IT environment. Whether creating reports for a specific performance metric or creating a report to show a consolidated view of an entity's capacity, Operations Manager provides a familiar set of tools to deliver reports on-demand or on a subscription basis.

Understanding how and when to create specific reports is a fundamental step toward leveraging Operations Manager data without overwhelming your customers with information. It is critical to know when the built-in reports from Operations Manager can be used and when to consider creating custom reports that provide exactly what customers want.

Using built-in reports

Most of the management packs available for Operations Manager are deployed with reports. An extensive collection of reports are available to you upon installation of the Operations Manager Reporting server. You can access reports in the Reporting workspace in the Operations Manager console (see Figure 3-1). These reports provide a way to visualize all the collected data so you can monitor and make educated decisions about your environment. The built-in reports are helpful for understanding your environment on a deeper level.

Whether you’re new to reporting or have many years’ experience, the built-in reports can provide insight and guidance for creating your own custom reports. You can leverage tools such as Report Builder or SQL Server Data Tools to review the syntax and structure of the built-in reports. However, you should never edit a report provided by a vendor in a management pack; you should always download a copy of the report and edit the copy instead of the original.


Reports - OM_MGMT - Op
File
Edit
View
Go
Tasks
Tools
Help
Search
Open
Scope
Find
Tasks
Reporting
Reporting
Application Monitoring
Client Monitoring Views Library
Contoso Example Report Library
Microsoft Data Warehouse Reports
Microsoft Generic Report Library
Microsoft ODR Report Library
Microsoft Service Level Report Library
Microsoft SharePoint Foundation 201
Microsoft System Center Virtual Mach
Network Management Reports
Red Hat Enterprise Linux Server 6 Ope
SharePoint 2010 Site Collection Exten
SQL Server 2005 (Monitoring)
SQL Server 2008 (Monitoring)
SQL Server 2012 (Monitoring)
System Center Core Monitoring Repo
Web Application Availability Monitor
Windows Server 2003 Operating Syste
Monitoring
Authoring
Reporting
Reports
Name

Windows Server 2012 R2 Memory Performance History (Available MB)
Windows Server 2012 R2 Memory Performance History (Page Reads per Sec)
Windows Server 2012 R2 Memory Performance History (Page Writes per Sec)
Windows Server 2012 R2 Memory Performance History (Pages per Sec)
Windows Server 2012 R2 Operating System Configuration
Windows Server 2012 R2 Operating System Performance
Windows Server 2012 R2 Paging File Performance History (Percentage Usage)
Windows Server 2012 R2 Performance History
Windows Server 2012 R2 Performance History (Context Switches per Sec)
Windows Server 2012 R2 Performance History (Percent Interrupt Time)
Windows Server 2012 R2 Performance History (Percent Processor Time)
Windows Server 2012 R2 Performance History (Processor Queue Length)
Windows Server 2012 R2 Physical Disk Performance History (Avg Disk Queue Length)
Windows Server 2012 R2 Pool Performance History (Nonpaged Bytes)
Windows Server 2012 R2 Pool Performance History (Paged Bytes)

Report Details:

The Performance History report displays one chart with these performance items:

- Total Processor % Processor Time

FIGURE 3-1 The Reporting workspace in the Operations Manager console

Most Operations Manager reports can be described as belonging to one of the following categories:

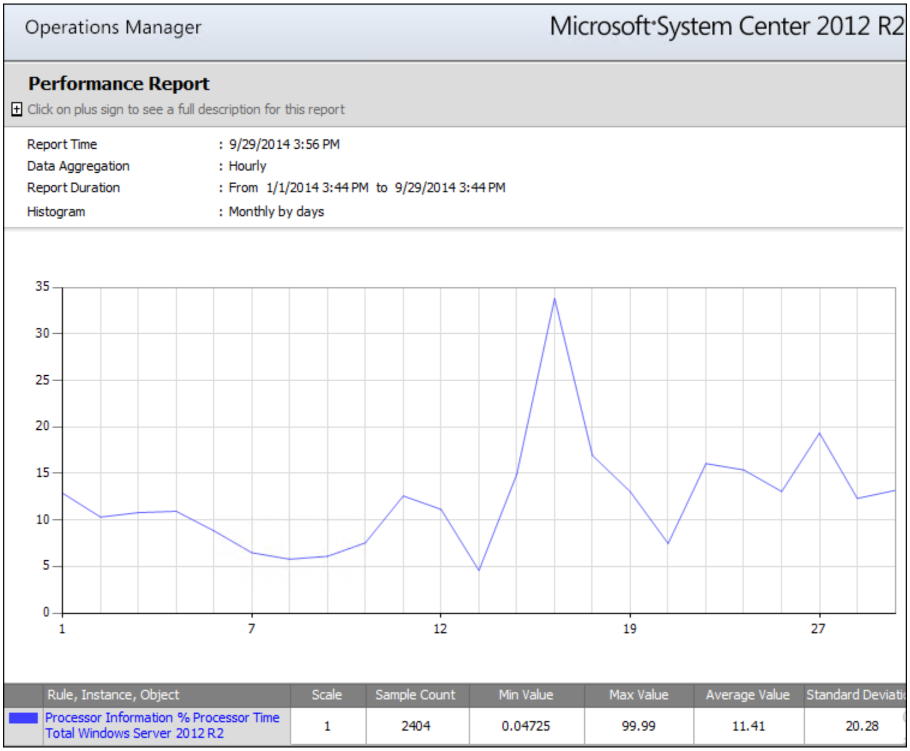
- **Performance report** Displays performance counters graphically based on the objects that are provided for the report
- **Performance detail report** Displays performance counters graphically but also shows minimum, maximum, and standard deviation of the counters
- **Configuration report** Displays configuration data and how configuration has changed over time
- **Availability report** Shows the health of objects as it relates to uptime and downtime and whether the health is in a critical, warning, or healthy state
- **Service level report** Shows service levels and how the objects being reported on follow those service levels

While the previous list of the different types of reports is by no means exhaustive, it does cover the majority of the reports you will use or create as an Operations Manager administrator.

Using custom reports

There are several good reasons to create your own reports. Internal or external customers may request reports that appear different than the built-in ones, they may want additional or less data on a report, or they may ask for a company-branded report on custom business applications and service monitoring. All of these reasons can justify creating your own reports; however, it is imperative to remember that custom reporting can be time consuming, so it is important to make sure that everyone understands both the cost and the value of the customizations needed. The following example illustrates the tradeoffs involved when deciding whether to customize a report to address a specific customer's needs:

Josh Pollock is an administrator for Litware, Inc. He is the Operations Manager administrator for his company. One of his internal customers reaches out to him about the performance report he gets for his servers (shown in Figure 3-2). The customer prefers a bar graph to the line graph used in the standard report. Josh is uncomfortable with this request, but he wants to make his internal customer happy, so he spends two weeks scouring TechNet forums and blogs to find out how to change this report.



Operations Manager

Microsoft System Center 2012 R2

Performance Report

⊞ Click on plus sign to see a full description for this report

Report Time	: 9/29/2014 3:56 PM
Data Aggregation	: Hourly
Report Duration	: From 1/1/2014 3:44 PM to 9/29/2014 3:44 PM
Histogram	: Monthly by days

	Rule, Instance, Object	Scale	Sample Count	Min Value	Max Value	Average Value	Standard Deviati
🟦	Processor Information % Processor Time Total Windows Server 2012 R2	1	2404	0.04725	99.99	11.41	20.28

FIGURE 3-2 A performance report for the Windows Server 2012 R2 computer group

In this scenario, do you think changing the report was worth Josh's effort? Two weeks to customize this report is quite a lot of time. When considering custom reporting, be sure to weigh the level of effort needed to create the custom report against the urgency of the requirement. The hours Josh worked to change the report's graphs equate to money that his time cost. Ultimately, whether the level of effort to change the graph is worth the cost comes down to a business decision.

Duplicating existing reports

If you must change an existing report, it is best to copy the .rdl file from SQL Server Reporting Services (SSRS) and duplicate it in another folder on the report server. Follow these steps to copy and duplicate an .rdl file:

1. Navigate to the Report Manager of your SSRS server by typing its URL in the address bar of a Web browser. By default, the URL is http://<reportservername>/reports.
2. Select a report to duplicate. Most likely your report will be in the Microsoft.SystemCenter.DataWarehouse.Report.Library folder where the generic reports that are linked to by other reports are stored.
3. Click Details View on the ribbon, click the down arrow to the right of the report you want to duplicate, and then click Manage.
4. On the Report Properties screen, click Download. This saves the .rdl file on your computer. If the report has a corresponding .rpdl file, download that file as well.
5. Navigate to the folder where you want to upload the report and click Upload File on the ribbon.
6. Upload the .rdl and .rpdl files to SSRS and use your favorite report authoring tool (Report Builder or SSDT) to edit the report as necessary.

NOTE **It is not advised to modify a built-in report from any management pack. The recommended practice is to copy the report file, which is an XML file with the report definition language extension (.rdl), from its original folder within SQL Server Reporting Services to another folder before making any changes. Any modifications to the built-in reports can be lost if the management pack is upgraded.**

Creating custom reports

Occasionally, business needs demand a custom report that contains more specific data and looks different than any of the existing reports. In this case, it is more feasible to create a new report rather than use a copy of an existing report To create a new report customized to meet your business requirements and save it in a new management pack, follow the steps outlined in Chapter 6, "Authoring reports in SQL Server Data Tools," and Chapter 7, "Bringing it all together into a management pack."

Examples of reporting scenarios

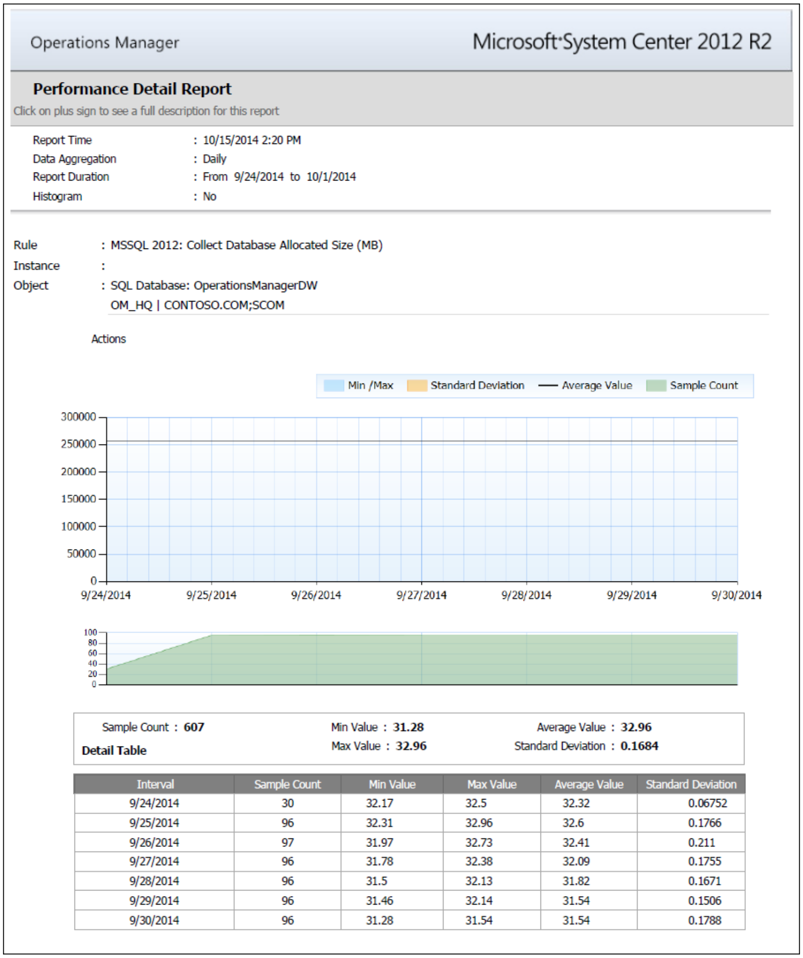
Many reporting scenarios are developed in response to the information gathered in the planning stage, as detailed in the “Planning reports” section of Chapter 2, “Operations Manager reporting basics.” Each report requires a clear understanding of how the information is going to be used. Often, however, reports are created without a clearly defined purpose. Whether the report is being delivered to a system administrator or a business executive, you should always know what information is being presented and how it's going to be used. This may appear to be an obvious consideration, but because many reports are requested without a defined purpose or target audience, it’s worth examining closer. To demonstrate the benefit of creating Operations Manager reports that meet specific, defined needs, the sections that follow provide several examples of different scenarios and the kinds of reports that they typically require.

Database administrator

Nancy Anderson is a database administrator for Litware, Inc. She has requested a daily report to monitor the growth of the Operations Manager data warehouse database. Nancy is concerned about exceeding the storage capacity of the clustered SQL Server instance hosting the Operations Manager data warehouse database. Although the SQL Server Management Pack has been configured and tuned to monitor the health, state, and performance of the SQL Server instance, she would like a report delivered each Monday at 7:00 a.m. in an easy-to-read format.

In this scenario, your target audience is primarily technical staff. The purpose of the report is to proactively monitor the growth of the Operations Manager data warehouse database to measure storage consumption and to forecast any additional storage allocation needs. Since you are creating a report with a scheduled weekly delivery to a specific email address, there’s no need to provide drill-down capabilities. Nancy’s only directive about the format of the report is that she can quickly and easily read it to determine the storage capacity. The final decision about the format is up to you. Creating a subscription schedule allows you to monitor the system resource usage during the report rendering to determine if any performance enhancement techniques are needed.

You have a general idea of the information requested, how it will be used, at what frequency, and by whom. At a minimum, this is enough to ensure that the report created will solve a specific problem; Nancy has provided enough information to determine the reporting parameters needed to create the report. You use this information to create a performance detail report based on the MSSQL 2012: Collect Database Total Free Space (MB) rule for the OperationsManagerDW database (see Figure 3-3).



Performance Detail Report

Click on plus sign to see a full description for this report

Report Time	: 10/15/2014 2:20 PM
Data Aggregation	: Daily
Report Duration	: From 9/24/2014 to 10/1/2014
Histogram	: No

Rule	: MSSQL 2012: Collect Database Allocated Size (MB)
Instance	:
Object	: SQL Database: OperationsManagerDW OM_HQ | CONTOSO.COM;SCOM

Sample Count : **607** Min Value : **31.28** Average Value : **32.96**

Max Value : **32.96** Standard Deviation : **0.1684**

Detail Table

Interval	Sample Count	Min Value	Max Value	Average Value	Standard Deviation
9/24/2014	30	32.17	32.5	32.32	0.06752
9/25/2014	96	32.31	32.96	32.6	0.1766
9/26/2014	97	31.97	32.73	32.41	0.211
9/27/2014	96	31.78	32.38	32.09	0.1755
9/28/2014	96	31.5	32.13	31.82	0.1671
9/29/2014	96	31.46	32.14	31.54	0.1506
9/30/2014	96	31.28	31.54	31.54	0.1788

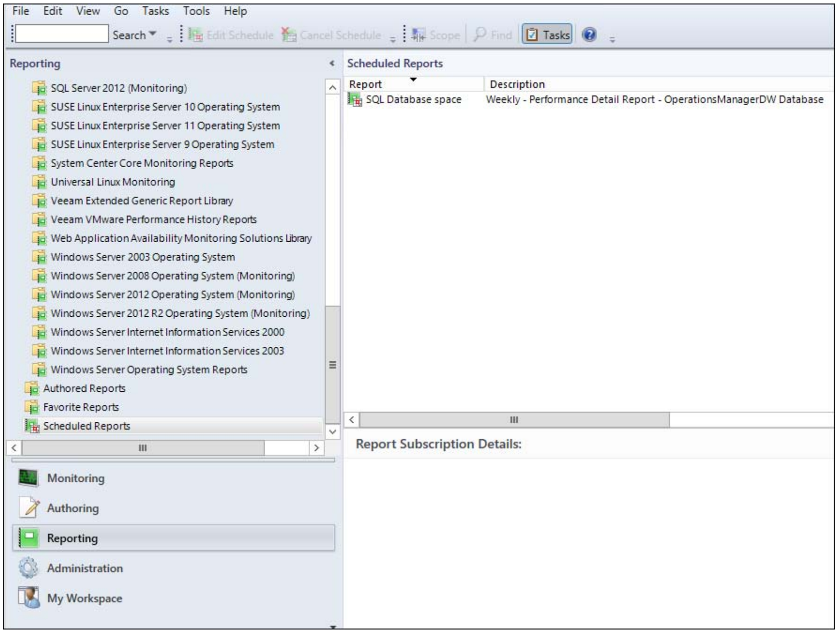
FIGURE 3-3 A performance detail report for the MSSQL 2012 Collect Database Total Free Space (MB) rule

After the report is created, you can determine how it will be delivered based on the information you gathered before the report's creation. You know that Nancy wants the report delivered to her email address on a daily basis. However, she didn't specify a preference for the report format other than it should be easy to read. With this knowledge, you can create a subscription schedule to automate the delivery of the report. After rendering the report, you can schedule the report to be sent via email delivery to one or more email addresses or email distribution lists, specify the rendered format to an XML Paper Specification or .xps file format, and configure the daily schedule of the file at a specific time. If needed, you can configure the report schedule to end at a specific date.

The following list details the criteria used to schedule the report delivery settings:

- **Delivery Method** E-mail
- **To** name@litware.com
- **Render Format** PDF
- **Subject** MSSQL 2012 Database Free Space Performance Detail Report
- **Schedule** Weekly

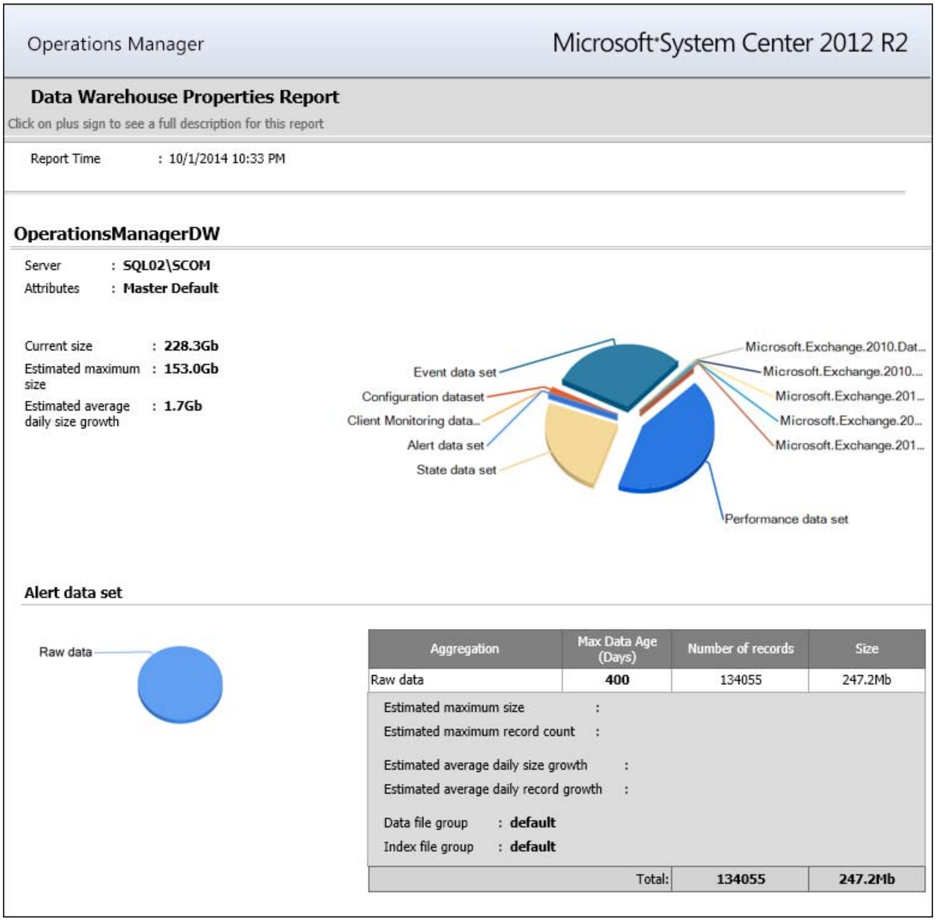
Figure 3-4 displays the report subscription as shown in the Scheduled Reports view of the Reporting workspace.


File
Edit
View
Go
Tasks
Tools
Help
Search
Edit Schedule
Cancel Schedule
Scope
Find
Tasks
Reporting
SQL Server 2012 (Monitoring)
SUSE Linux Enterprise Server 10 Operating System
SUSE Linux Enterprise Server 11 Operating System
SUSE Linux Enterprise Server 9 Operating System
System Center Core Monitoring Reports
Universal Linux Monitoring
Veeam Extended Generic Report Library
Veeam VMware Performance History Reports
Web Application Availability Monitoring Solutions Library
Windows Server 2003 Operating System
Windows Server 2008 Operating System (Monitoring)
Windows Server 2012 Operating System (Monitoring)
Windows Server 2012 R2 Operating System (Monitoring)
Windows Server Internet Information Services 2000
Windows Server Internet Information Services 2003
Windows Server Operating System Reports
Authored Reports
Favorite Reports
Scheduled Reports
Monitoring
Authoring
Reporting
Administration
My Workspace
Scheduled Reports
Report Subscription Details:

Report	Description
SQL Database space	Weekly - Performance Detail Report - OperationsManagerDW Database

FIGURE 3-4 The Reporting workspace showing the Scheduled Reports properties

As an alternative to creating the report from scratch, you can deliver more valuable insight to your database administrator by creating a scheduled delivery of the Data Warehouse Properties report located in the Microsoft Data Warehouse Reports folder. This is a built-in report provided to show the Data Warehouse properties and grooming settings (see Figure 3-5).



Operations Manager

Microsoft System Center 2012 R2

Data Warehouse Properties Report

Click on plus sign to see a full description for this report

Report Time : 10/1/2014 10:33 PM

OperationsManagerDW

Server : SQL02\SCOM

Attributes : Master Default

Current size : 228.3Gb

Estimated maximum size : 153.0Gb

Estimated average daily size growth : 1.7Gb

Event data set
Configuration dataset
Client Monitoring data...
Alert data set
State data set
Microsoft.Exchange.2010.Dat...
Microsoft.Exchange.2010....
Microsoft.Exchange.201...
Microsoft.Exchange.20...
Microsoft.Exchange.201...
Performance data set

Alert data set

Aggregation	Max Data Age (Days)	Number of records	Size
Raw data	400	134055	247.2Mb

Estimated maximum size :

Estimated maximum record count :

Estimated average daily size growth :

Estimated average daily record growth :

Data file group : default

Index file group : default

Total: 134055 247.2Mb

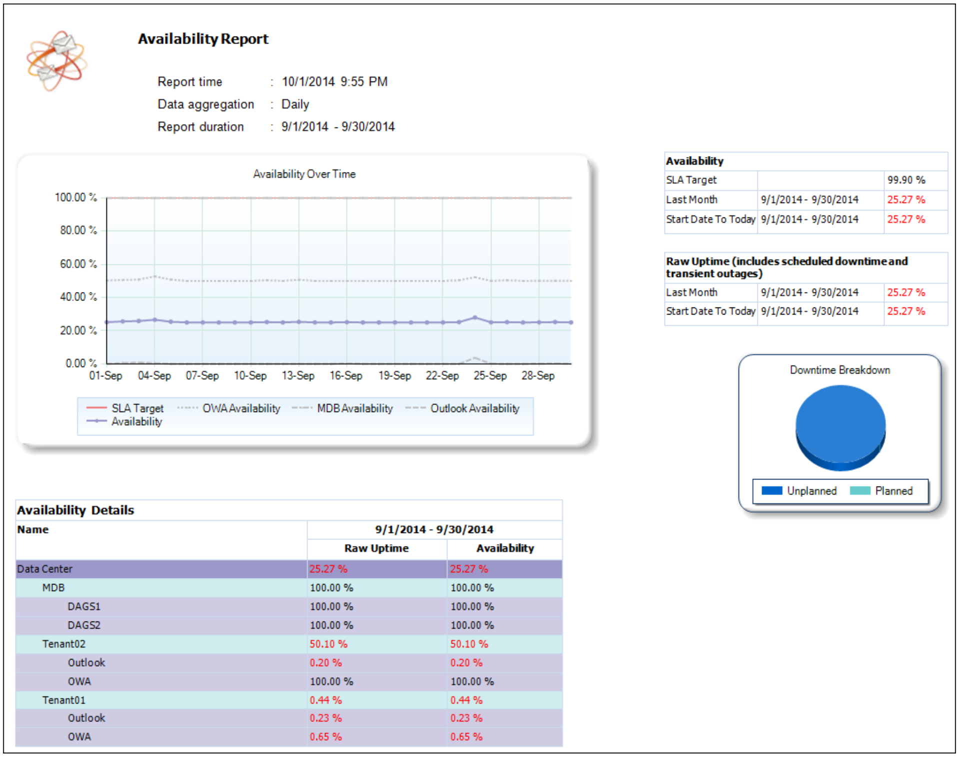
FIGURE 3-5 The Data Warehouse Properties report

Chief Information Officer

Andreas Herbinger is the CIO for Litware, Inc., an application hosting provider delivering email as a service to small businesses. Andreas has a customer satisfaction problem with one of his most valued customers, Contoso, Inc. Contoso has complained repeatedly about disruptions in service and slow response time in email delivery. He has asked you to provide him with a report detailing the customer's Microsoft Exchange 2010 availability to determine if any service level agreements (SLAs) were missed during the monthly billing cycle. Andreas wants the report delivered immediately to his email address in Microsoft Word format, and he wants to be able to drill down into the report sections as needed.

From this scenario, you determine that your target audience is executive leadership. Andreas has clearly defined the report parameters including the duration, rendering format, and drill-down requirements. As with the previous scenario, you are creating a report with a scheduled delivery to an email address. Most importantly, you must provide the capability to drill down into the report. You decide to create another subscription schedule to send the CIO the report and to include a link to the report on the Operations Manager Reporting server, enabling him to drill down into the report for more detailed information.

With the report parameters clearly defined, you can fulfill the CIO's request by immediately scheduling the delivery of the Microsoft Exchange 2010 Reports SLA report located in the Exchange 2010 folder. This is a built-in report provided to show the Microsoft Exchange 2010 SLA availability numbers (see Figure 3-6).



Availability Report

Report time : 10/1/2014 9:55 PM
Data aggregation : Daily
Report duration : 9/1/2014 - 9/30/2014

Availability		
SLA Target		99.90 %
Last Month	9/1/2014 - 9/30/2014	25.27 %
Start Date To Today	9/1/2014 - 9/30/2014	25.27 %

Raw Uptime (includes scheduled downtime and transient outages)		
Last Month	9/1/2014 - 9/30/2014	25.27 %
Start Date To Today	9/1/2014 - 9/30/2014	25.27 %

Availability Details

Name	**9/1/2014 - 9/30/2014**	
	Raw Uptime	**Availability**
Data Center	25.27 %	25.27 %
MDB	100.00 %	100.00 %
DAGS1	100.00 %	100.00 %
DAGS2	100.00 %	100.00 %
Tenant02	50.10 %	50.10 %
Outlook	0.20 %	0.20 %
OWA	100.00 %	100.00 %
Tenant01	0.44 %	0.44 %
Outlook	0.23 %	0.23 %
OWA	0.65 %	0.65 %

FIGURE 3-6 The Microsoft Exchange 2010 Reports SLA report

This page intentionally left blank

CHAPTER 4

Overview of report authoring tools

A number of tools extend the built-in capabilities of Microsoft System Center Operations Manager. Whether you choose a tool developed by Microsoft, Microsoft partners, or even by the community, as an Operations Manager administrator, you are sure to find a tool to accommodate your needs. A number of tools are available to create custom management packs for Operations Manager, but this chapter focuses on some of the more widely used tools for creating custom reports, including:

- Microsoft SQL Server Report Builder
- Microsoft SQL Server Data Tools
- Microsoft System Center 2012 Visual Studio Authoring Extensions
- Microsoft System Center Operations Manager 2007 R2 Authoring Console
- Microsoft Power View

See also *The report creation procedures in Operations Manager 2012 have not changed from Operations Manager 2007; therefore, you should refer to the Operations Manager 2007 Report Authoring Guide for guidance, located at http://technet.microsoft.com/en-us/library/hh528528.aspx.*

Report Builder

The Report Builder tool is automatically installed with SQL Server Reporting Services (SSRS). You can also download the standalone version of Report Builder from the following URLs:

- **Microsoft SQL Server 2008 R2** See *http://www.microsoft.com/en-us/download/details.aspx?id=6116*
- **Microsoft SQL Server 2012** See *http://www.microsoft.com/en-us/download/details.aspx?id=29072*

Before installing Report Builder, you must install the appropriate version of Microsoft .NET Framework: Microsoft .NET Framework 3.5.1 for SQL Server 2008 R2 and Microsoft .NET Framework 4.0 for SQL Server 2012. When SSRS is installed and configured, you can launch Report Builder from the SSRS Reports webpage by clicking on the Report Builder link at the top of the page, as shown in Figure 4-1. Alternatively, you can install Report Builder locally on your workstation from the download links provided above.

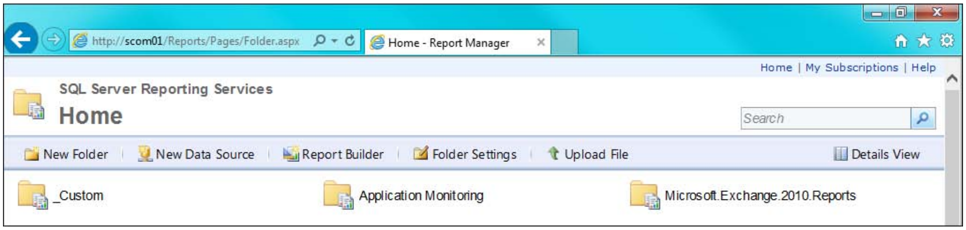


FIGURE 4-1 SSRS Reports webpage

> **TIP** You can also launch this application by navigating to the following URL in Internet Explorer on your SSRS server: http://scom01/ReportServer/ReportBuilder_3_0_0_0.application.

Report Builder includes a variety of visualization options to make your reports more visually appealing, such as maps, sparklines, gauges, indicators, charts, and databars. As shown in Figure 4-2, the Report Builder console looks similar to the Microsoft Office suite interface, featuring a ribbon to help you quickly find customization options. These options are logically grouped into tabs. The ribbon is where you manipulate text and table properties. The Insert tab is where you add data visualizations and images. On the View tab you can customize the Report Builder workspace by activating the Ruler, Report Data, Grouping, and Property panes. When your data sources and datasets are configured, you can create a report using a drag-and-drop operation. Chapter 5, "Authoring reports in Report Builder," guides you through creating a custom report using Report Builder.

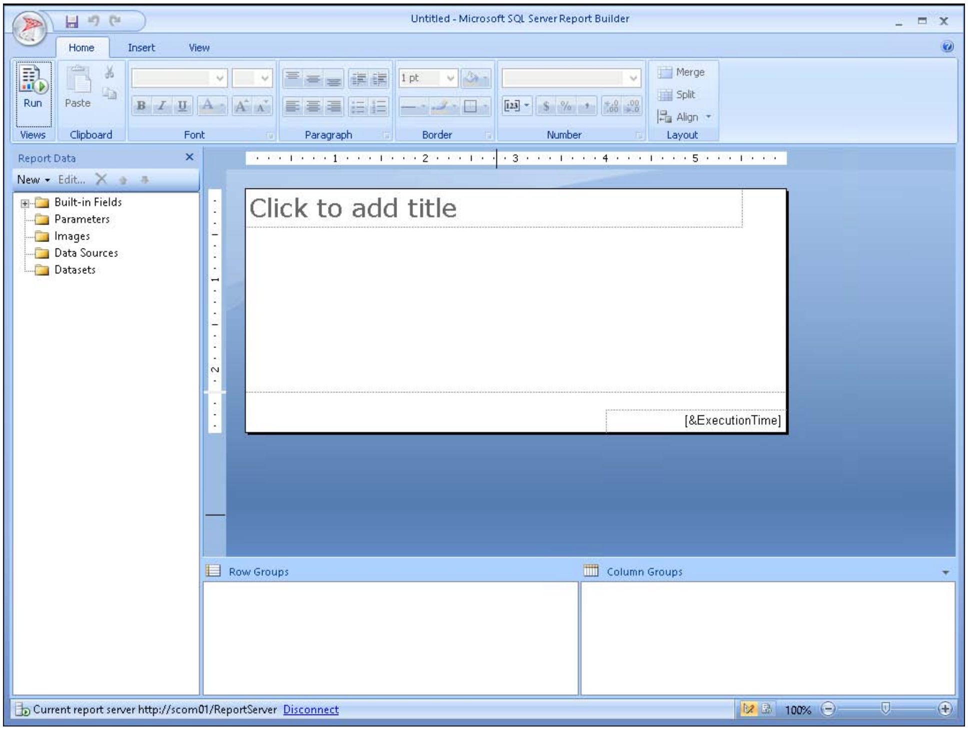


FIGURE 4-2 Report Builder console

See also *Refer to the following URL for more information about using Report Builder: http://technet.microsoft.com/en-us/library/ms159253%28v=sql.105%29.aspx.*

SQL Server Data Tools

Microsoft SQL Server Data Tools (SSDT), a Microsoft Silverlight feature available with Microsoft SQL Server 2012, contains the most recent version of Report Designer. In Microsoft SQL Server 2005 and Microsoft SQL Server 2008, SSDT was called Microsoft SQL Server Business Intelligence Development Studio (BIDS).

NOTE You can install SSDT directly from the SQL Server 2012 installation media or download it from *http://msdn.microsoft.com/en-us/data/hh297027*.

SSDT Report Designer and Report Builder 3.0 share a similar user interface and functionality, and you can achieve your desired results in a similar manner with either tool. However, SSDT Report Designer is preferable to Report Builder in a number of situations, including when you want to create multidimensional database queries in a graphical query designer, to create data mining queries, to use IntelliSense, or to take advantage of the tool's more comprehensive project features. Chapter 6 provides guidance for creating reports using SSDT.

NOTE SSDT will install the Visual Studio shell if Visual Studio is not already installed on your workstation.

See also *You can find a more complete comparison between the two products at http://technet.microsoft.com/en-us/library/dd207010%28v=sql.105%29.aspx.*

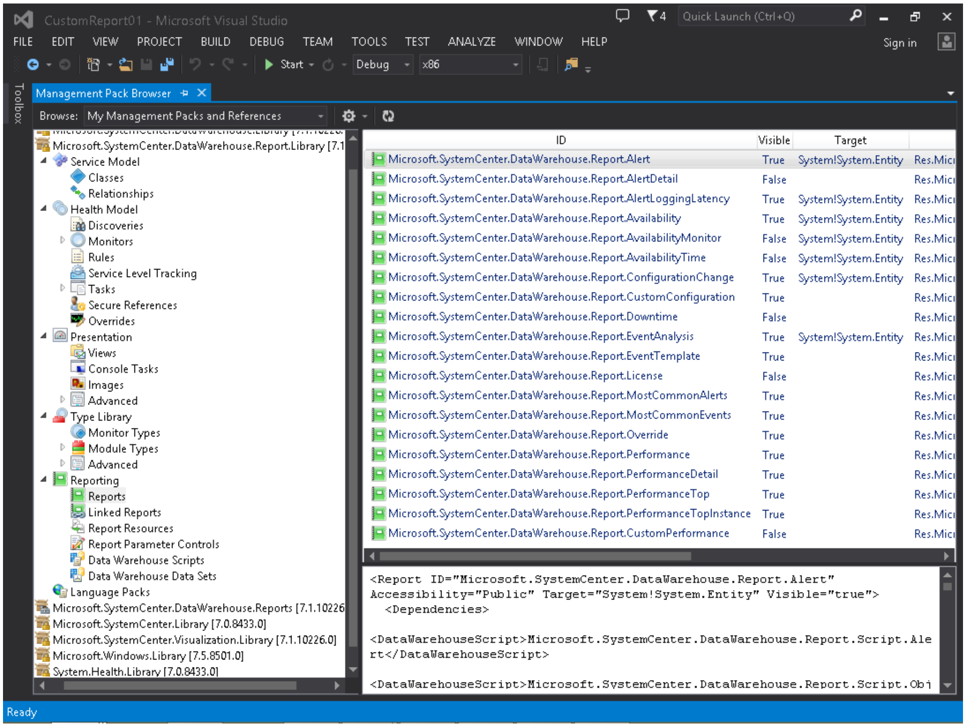
Visual Studio Authoring Extensions

The System Center 2012 Visual Studio Authoring Extensions (VSAE) is an add-in for Microsoft Visual Studio 2012 and Microsoft Visual Studio 2013. At the time of this writing, this add-in is supported only in Visual Studio Professional and Visual Studio Ultimate. VSAE features one of the more advanced methods for creating custom management packs and custom reports. It is the most complex tool for creating custom management packs and reports when compared to the other options covered in this book. Chapter 7, "Building management packs for reporting," provides guidance for creating a custom report using VSAE as well as additional resources for creating management packs in Visual Studio.

You can use VSAE to create custom management packs and reports for Operations Manager 2007 R2 and later. In VSAE, you work directly with the XML code of the management pack, which allows more granular control and configurations of the management pack. A great feature of VSAE is the Management Pack Browser, which allows you to view the entire service model of a management pack. Figure 4-3 shows the service model of the Microsoft System Center Data Warehouse Reports Library.

VSAE provides additional tools for managing management packs, such as the Management Pack Simulator, Management Pack Best Practice Analyzer, and others. Figure 4-3 displays the Microsoft System Center Data Warehouse Report Library management pack view within the Management Pack Browser. The Management Pack Browser displays a breakdown of each element within the selected management pack. When Reports is selected, all available reports within the selected management pack appear in a list. Click a specific report and the XML code appears in the bottom pane. This is helpful if you are curious about how a particular report works or if you want to reuse existing code in custom management pack or report files. The same applies for any of the other elements within the Management Pack Browser. As one of the more advanced and complex authoring tools, VSAE gives you complete control over your management packs and reports.

IMPORTANT It is important to point out that editing a sealed management pack from a vendor is not supported.


CustomReport01 - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Sign in
Start
Debug
x86
Toolbox
Management Pack Browser
Browse: My Management Packs and References
Microsoft.SystemCenter.DataWarehouse.Report.Library [7.1
Service Model
Classes
Relationships
Health Model
Discoveries
Monitors
Rules
Service Level Tracking
Tasks
Secure References
Overrides
Presentation
Views
Console Tasks
Images
Advanced
Type Library
Monitor Types
Module Types
Advanced
Reporting
Reports
Linked Reports
Report Resources
Report Parameter Controls
Data Warehouse Scripts
Data Warehouse Data Sets
Language Packs
Microsoft.SystemCenter.DataWarehouse.Reports [7.1.10226
Microsoft.SystemCenter.Library [7.0.8433.0]
Microsoft.SystemCenter.Visualization.Library [7.1.10226.0]
Microsoft.Windows.Library [7.5.8501.0]
System.Health.Library [7.0.8433.0]
Ready

ID	Visible	Target	
Microsoft.SystemCenter.DataWarehouse.Report.Alert	True	System!System.Entity	Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.AlertDetail	False		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.AlertLoggingLatency	True	System!System.Entity	Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.Availability	True	System!System.Entity	Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.AvailabilityMonitor	False	System!System.Entity	Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.AvailabilityTime	False	System!System.Entity	Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.ConfigurationChange	True	System!System.Entity	Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.CustomConfiguration	True		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.Downtime	False		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.EventAnalysis	True	System!System.Entity	Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.EventTemplate	True		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.License	False		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.MostCommonAlerts	True		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.MostCommonEvents	True		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.Override	True		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.Performance	True		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.PerformanceDetail	True		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.PerformanceTop	True		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.PerformanceTopInstance	True		Res.Mici
Microsoft.SystemCenter.DataWarehouse.Report.CustomPerformance	False		Res.Mici

```
<Report ID="Microsoft.SystemCenter.DataWarehouse.Report.Alert"
Accessibility="Public" Target="System!System.Entity" Visible="true">
  <Dependencies>

<DataWarehouseScript>Microsoft.SystemCenter.DataWarehouse.Report.Script.Ale
rt</DataWarehouseScript>

<DataWarehouseScript>Microsoft.SystemCenter.DataWarehouse.Report.Script.Obj
```

FIGURE 4-3 VSAE Management Pack Browser with Reports selected

Operations Manager 2007 R2 Authoring Console

The Operations Manager 2007 R2 Authoring Console is an excellent tool for creating custom management packs and reports. The Authoring Console is included in the System Center Operations Manager 2007 R2 Authoring Resource Kit, which is available at *http://www.microsoft.com/en-us/download/details.aspx?id=18222*. This resource kit contains a number of VSAE tools. You may also notice that the Authoring Console interface is similar to that of the Operations Manager Operations console.

The Authoring Console remains a supported tool for Operations Manager 2012 management pack and report authoring. It is similar to the Authoring workspace of the Operations Manager Operations console. If you are familiar with the Operations console, in the Authoring Console you will recognize similar wizards for creating various rules, monitors, overrides, groups, tasks, views, and other items. The Authoring Console exposes more advanced features, which will be explained in further detail in Chapter 7.

NOTE **Because the Authoring Console was developed for Operations Manager 2007 R2 and has not been updated since that version, you may run into some compatibility issues while using the tool to edit Operations Manager 2012 management packs. The Authoring Console is not aware of the Operations Manager 2012 management pack structure or names and therefore may produce errors during your management pack or report creation efforts. Refer to the Management Pack Authoring section of the release notes for System Center 2012 Operations Manager located at *http://technet.microsoft.com/en-us/library/jj656656.aspx*.**

Power View

Microsoft Power View is a new business intelligence (BI) feature introduced in Microsoft Office 2013 that allows you to retrieve data from various tabular data models and create rich, interactive dashboards. Power View is available as a feature in Microsoft Excel and can also be used in SharePoint Server 2010 or SharePoint Server 2013. You can use either option to connect to your Operations Manager databases and produce intuitive dashboards for your customers.

See also *For installation guidance on configuring your SharePoint environment for using Power View, refer to http://msdn.microsoft.com/en-us/library/hh231687.aspx.*

Most reporting options require a moderate to advanced level of effort to create custom reports. Power View makes custom report creation much easier to learn and use in comparison to some of the other report authoring tools that are discussed in this book. Power View works similarly to PivotTables. After creating a data model, select the data you want to include in your report, and then it's mostly a drag-and-drop process to create reports. Your end users can create their own reports using various visualizations and reporting features, such as graphs, charts, key performance indicators (KPIs), drill downs, and maps. There are a number of options for creating reports, all within a familiar user interface. Chapter 8, "Authoring dashboards in Power View," focuses in more detail on creating custom dashboards using Power View.

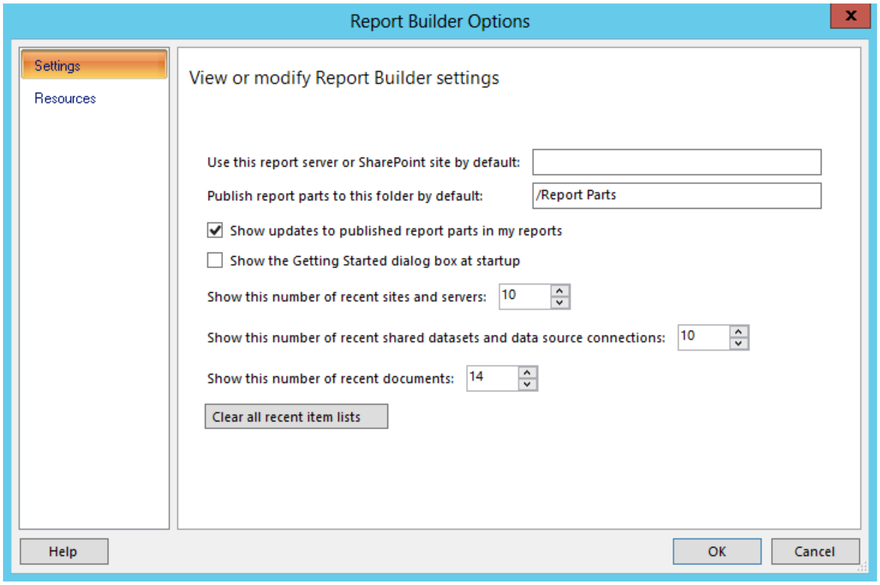
CHAPTER 5

Authoring reports in Report Builder

This chapter discusses using Report Builder to extend Operations Manager reporting and creating custom reports. Topics covered include how to configure Report Builder and how to use it to create custom reports for your Operations Manager environment.

Configuring Report Builder for Operations Manager reports

Before you can create reports with Report Builder, you must configure it to connect to your SQL Server Reporting Services (SSRS) instance. To do so, open Report Builder, close the Getting Started dialog box that appears, and then click Options at the bottom of the Report Builder menu. This displays the Report Builder Options dialog box, shown in Figure 5-1.



Report Builder Options

View or modify Report Builder settings

Use this report server or SharePoint site by default:

Publish report parts to this folder by default: /Report Parts

☑ Show updates to published report parts in my reports

□ Show the Getting Started dialog box at startup

Show this number of recent sites and servers: 10

Show this number of recent shared datasets and data source connections: 10

Show this number of recent documents: 14

FIGURE 5-1 Report Builder Options dialog box

In the Report Builder Options dialog box, define the default SSRS site for Report Builder to use and the default folder for published reports. In this dialog box, you can also set the default number of recent items to show and clear the history. To simplify custom report creation, set the default SSRS page to your Operations Manager SSRS report page and set the default folder to the one where you want to publish the reports.

Creating a report

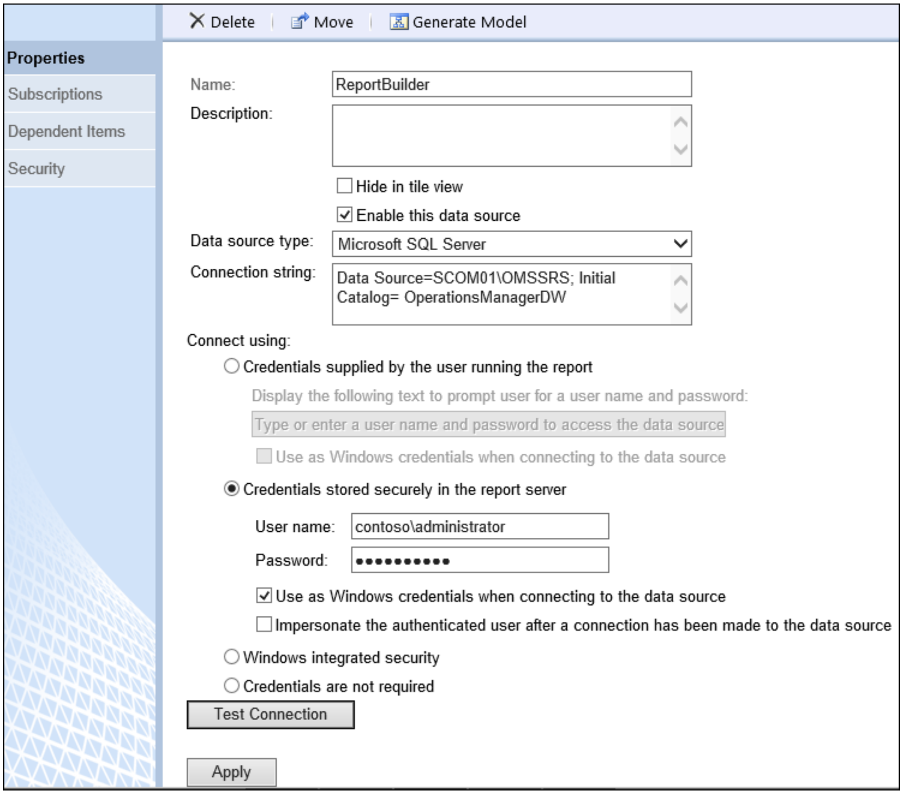
After you configure Report Builder, you can create a report. To illustrate how to build a report, this section steps through an example of creating a report showing processor usage data for the last seven days.

Configuring a data source

Before you create a report in Report Builder, you need to create a data source for pulling the report data. As a best practice, create a specific data source for this instead of using the default DataWarehouseMain data source installed by Operations Manager. Doing so enables you to control the permissions to the custom reports outside of the Operations Manager security model. This means you can allow other users to publish and run reports without adding these users to Operations Manager.

To create a data source, open Internet Explorer and navigate to the Operations Manager SSRS webpage. To get the correct URL, open the Operations Manager console, navigate to the Administration pane, select Settings, and double-click the Reporting item. Typically, the URL is http://<servername>/ReportServer.

On the reporting webpage, click Details View, and then click New Data Source. This will open the New Data Source page, similar to the one shown in Figure 5-2.



Name: ReportBuilder

Description:

□ Hide in tile view

☑ Enable this data source

Data source type: Microsoft SQL Server

Connection string: Data Source=SCOM01\OMSSRS; Initial Catalog= OperationsManagerDW

Connect using:

○ Credentials supplied by the user running the report

Display the following text to prompt user for a user name and password:

Type or enter a user name and password to access the data source

□ Use as Windows credentials when connecting to the data source

● Credentials stored securely in the report server

User name: contoso\administrator

Password: ●●●●●●●●●●●

☑ Use as Windows credentials when connecting to the data source

□ Impersonate the authenticated user after a connection has been made to the data source

○ Windows integrated security

○ Credentials are not required

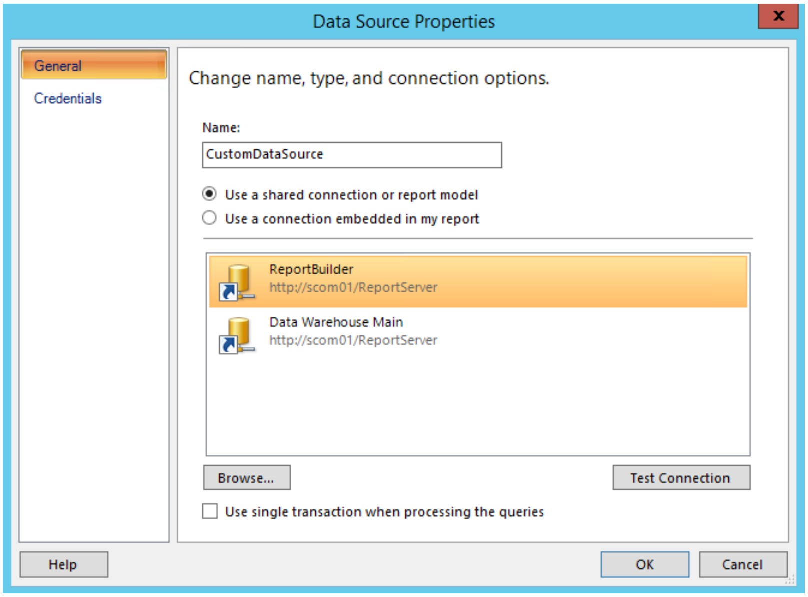
FIGURE 5-2 New Data Source creation page

Give the data source a name and description, and add the connection string. Select the Credentials Stored Securely In The Report Server option and add a user with read permissions on the data warehouse database. When finished, click Apply.

Adding the data source to a new report

The next step is to add the data source to a new report. In Report Builder, click the Ribbon button in the upper-right corner and choose New, Blank Report. A blank report opens for editing. This window is called the Report Design Window and was shown earlier as Figure 4-2 in the previous chapter.

To add the new data source to the report, right-click the Data Sources folder in the Report Data pane on the left side of the Report Design Window and click Add Data Source in the context menu that appears. The Data Source Properties dialog box shown in Figure 5-3 opens.



Change name, type, and connection options.

Name: CustomDataSource

● Use a shared connection or report model

○ Use a connection embedded in my report

ReportBuilder
http://scom01/ReportServer

Data Warehouse Main
http://scom01/ReportServer

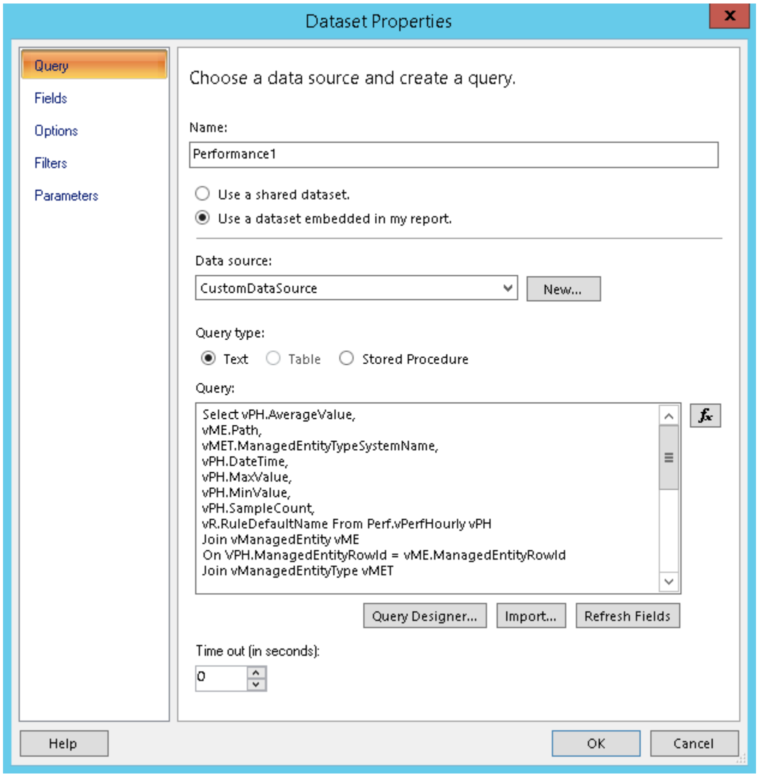
□ Use single transaction when processing the queries

FIGURE 5-3 Data Source Properties dialog box

Give the data source a name and select Use A Shared Connection Or Report Model. Click Browse to open a window listing the available items on the SSRS page. Select the data source you created earlier, and click Open. Click Test Connection to verify you can connect to SSRS, and then click OK. The data source appears in the Report Data Pane under the folder named Data Sources.

Creating a dataset for the report

With the data source added, you next need to create a dataset to provide the fields for the report. Right-click datasets in the Report Data Pane and click Add Dataset, which opens the Dataset Properties dialog box shown in Figure 5-4.



Choose a data source and create a query.

Name: Performance1

○ Use a shared dataset.
● Use a dataset embedded in my report.

Data source: CustomDataSource

Query type: ● Text ○ Table ○ Stored Procedure

Query:

```
Select vPH.AverageValue,
vME.Path,
vMET.ManagedEntityTypeSystemName,
vPH.DateTime,
vPH.MaxValue,
vPH.MinValue,
vPH.SampleCount,
vR.RuleDefaultName From Perf.vPerfHourly vPH
Join vManagedEntity vME
On VPH.ManagedEntityRowId = vME.ManagedEntityRowId
Join vManagedEntityType vMET
```

Time out (in seconds): 0

FIGURE 5-4 Dataset Properties dialog box

Give the dataset a name and select Use A Dataset Embedded In My Report. Select your data source from the Data Source list, and then select Text under Query Type.

Specifying the query for the report

At this point in the process, you could click Query Designer to build your query, and you might do so for advanced reports, but for simple reports like the example in this chapter, you can type your query in the Query textbox.

TIP **As a best practice, always target views in your query and not the data tables directly. Dataset data, such as Performance, State, Event, and Alert, are stored in partition tables. The views are automatically updated to retrieve data from the active partitions.**

Below is a simple Select query targeted at the performance hourly view to pull performance data:

```
Select
vME.Path,
vMET.ManagedEntityTypeSystemName,
vPH.DateTime,
vPH.AverageValue,
vPH.MaxValue,
vPH.MinValue,
vPH.SampleCount,
vR.RuleDefaultName From Perf.vPerfHourly vPH
Join vManagedEntity vME
On VPH.ManagedEntityRowId = vME.ManagedEntityRowId
Join vManagedEntityType vMET
On vME.ManagedEntityTypeRowId = vMET.ManagedEntityTypeRowId
Join vPerformanceRuleInstance vPRI
On vPH.PerformanceRuleInstanceRowId = vPRI.PerformanceRuleInstanceRowId
Join vRule vR
On vPRI.RuleRowId = vR.RuleRowId
Where
vMET.ManagedEntityTypeSystemName = 'Microsoft.Windows.OperatingSystem'
And
vR.RuleDefaultName = 'Processor % Processor Time Total Windows Server 2012'
And
vPH.DateTime >= DATEADD(Day, -7, GetDate())
```

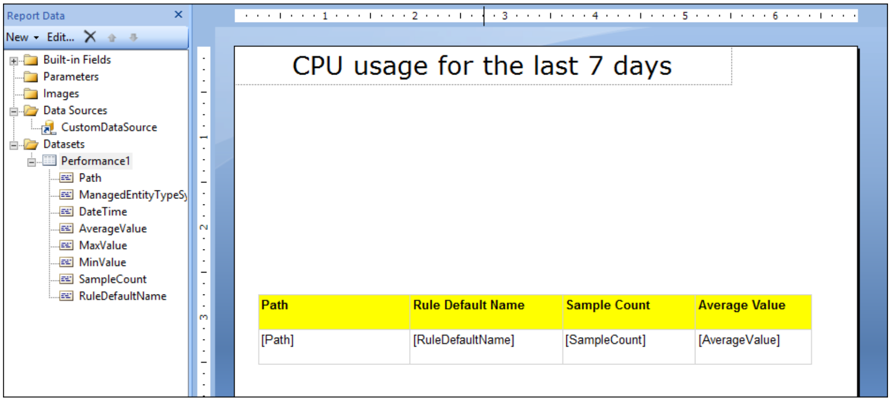
COMPANION CONTENT **The sample Select query is also provided in the companion files for this book.**

When the query is added, click OK, and you will be prompted to enter the data source credentials. After entering the credentials, the fields that are created by the dataset appear in the Report Data Pane under the new dataset. You can now use these fields when creating your report.

Designing the report

In the report design window, click the title box and type a title for the report. Try to be as specific as possible. After adding the title, click Insert, and then click Table, Insert Table. Click inside the design window and drag the table to the size desired.

Next, drag each field to a column in the table. For example, the report in this example includes four fields: Path, Rule Default Name, Sample Count, and Average Value. When done, the report should look like Figure 5-5.



CPU usage for the last 7 days

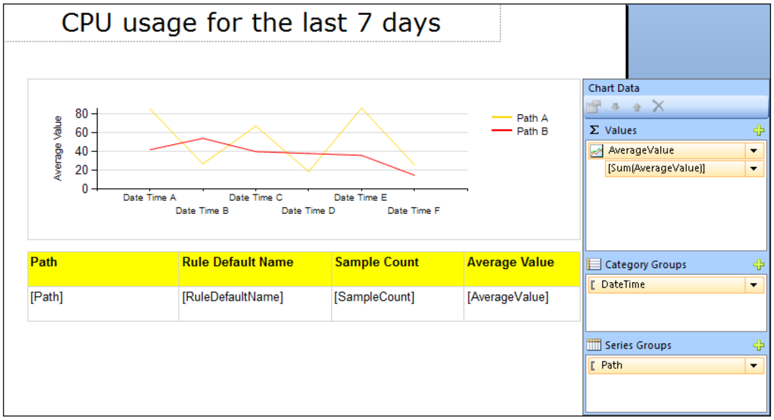
Path	Rule Default Name	Sample Count	Average Value
[Path]	[RuleDefaultName]	[SampleCount]	[AverageValue]

FIGURE 5-5 Report with title and table added

In Figure 5-5, each of the headers are highlighted in yellow so the column headers stand out in the finished report. To highlight the headers, right-click the text box, click Text Box Properties, and select a fill color.

To add a chart to the report, click Insert,on the Ribbon Menu and then click Chart, Insert Chart. Click in the design window and a Select Chart Type dialog box opens. Select Line as the chart type, and then click OK. Drag the chart where you want it and size it to your preferences. For this example, make the chart the same width as the table.

To associate the chart with your dataset, right-click the chart, and click Chart, Chart Properties. Select your dataset from the drop-down list. You can also select a color palette in the Chart Properties dialog box to change the colors in the chart. The report in Figure 5-6 uses the Fire palette. Make all selections and then click OK.



CPU usage for the last 7 days

Path	Rule Default Name	Sample Count	Average Value
[Path]	[RuleDefaultName]	[SampleCount]	[AverageValue]

Values: AverageValue, [Sum(AverageValue)]

Category Groups: [DateTime

Series Groups: [Path

FIGURE 5-6 Final report design

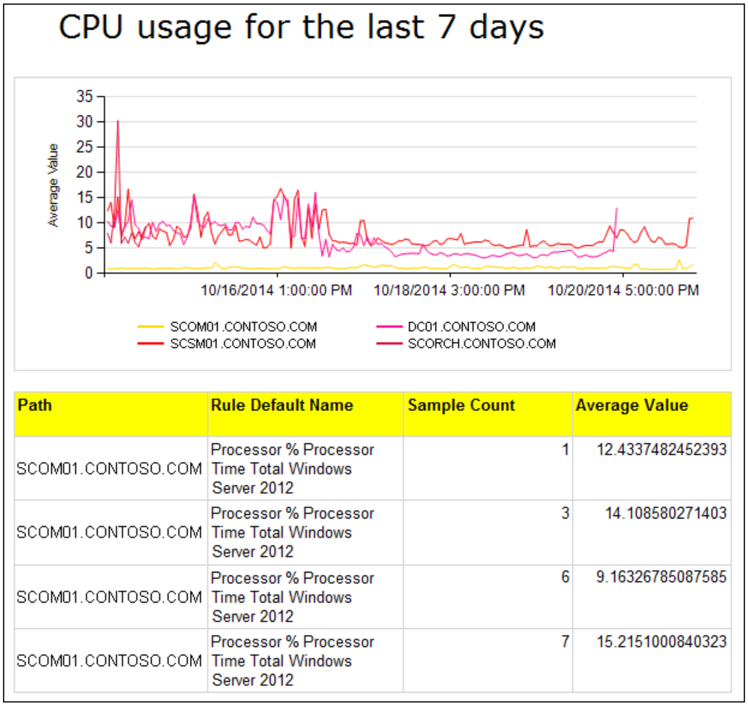
To define the axis and line titles, right-click each one and click Axis Title Properties. In the Axis Title Properties dialog box, you can either define a title or click the function symbol to assign a title dynamically. In the example report, name the vertical axis title Average Value. For the horizontal axis title, select the function symbol and click Fields, then double-click DateTime.

For the column values, you define the Category Groups and Values sections. The category groups are the horizontal axis values, and values correspond to the vertical axis. Click the plus sign next to Values and select AverageValue, and then click the plus sign next to Category Groups and select DateTime.

For the legend you need to add a series group, so click the plus sign next to Series Groups and select Path. Once this is completed, your report design should look like Figure 5-7.

Testing the report

With the report settings finalized, the report is ready to test and publish. You can also customize the table and chart to add different colors and borders, which can be easier to test if you add the properties window. To do this, click View, on the Ribbon Menu and select Properties. With this setting, when you select anything in the design window, the object's properties appear to the right of it. Figure 5-7 shows the final run report with a border and color change on the chart.



CPU usage for the last 7 days

Path	Rule Default Name	Sample Count	Average Value
SCOM01.CONTOSO.COM	Processor % Processor Time Total Windows Server 2012	1	12.4337482452393
SCOM01.CONTOSO.COM	Processor % Processor Time Total Windows Server 2012	3	14.108580271403
SCOM01.CONTOSO.COM	Processor % Processor Time Total Windows Server 2012	6	9.16326785087585
SCOM01.CONTOSO.COM	Processor % Processor Time Total Windows Server 2012	7	15.2151000840323

FIGURE 5-7 Image of the final report

CHAPTER 6

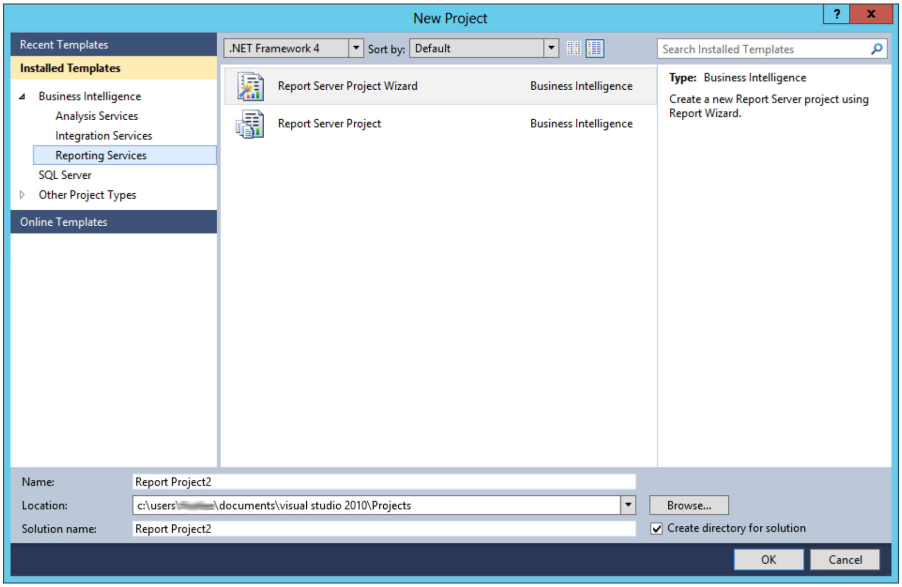
Authoring reports in SQL Server Data Tools

SQL Server Data Tools, commonly referred to as SSDT, is used to create custom reports for Operations Manager. Similarly to Chapter 5, "Authoring reports in Report Builder," this chapter uses an extended example to show how you can use SSDT to create custom reports and provides some example queries to retrieve data from the data warehouse.

Getting started with SSDT

You can use SSDT to create SQL projects for analysis services, integration services, reporting services, and much more. In fact, the initial SSDT screen reveals that the tool is designed for more than simply creating reports. However, when you move past the initial start screens and into the report design screen, the similarities between SSDT and Report Builder become more obvious. For the purposes of this book, this chapter focuses strictly on reporting services in SSDT.

To create a new project, open SSDT and click the New Project menu item. A New Project dialog box similar to the one shown in Figure 6-1 opens.



Report Server Project Wizard — Business Intelligence
Report Server Project — Business Intelligence

Type: Business Intelligence
Create a new Report Server project using Report Wizard.

Name: Report Project2
Location: c:\users\[redacted]\documents\visual studio 2010\Projects
Solution name: Report Project2
☑ Create directory for solution

FIGURE 6-1 New Project dialog box

Click Report Server Project and type a name for the project in the Name text box. Choose a save location and type a name for the solution in the Solution Name text box. When all the fields are complete, click OK. The design screen opens.

The design screen in SSDT is similar to the one in Report Builder, except the Solution Explorer pane is located on the right. Figure 6-2 shows the SSDT Solution Explorer.

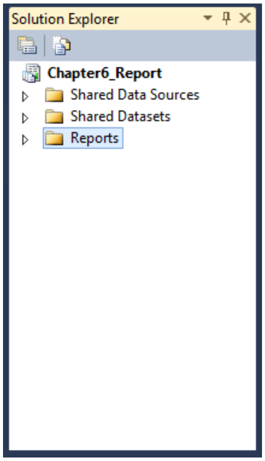


FIGURE 6-2 SSDT Solution Explorer

Creating a report

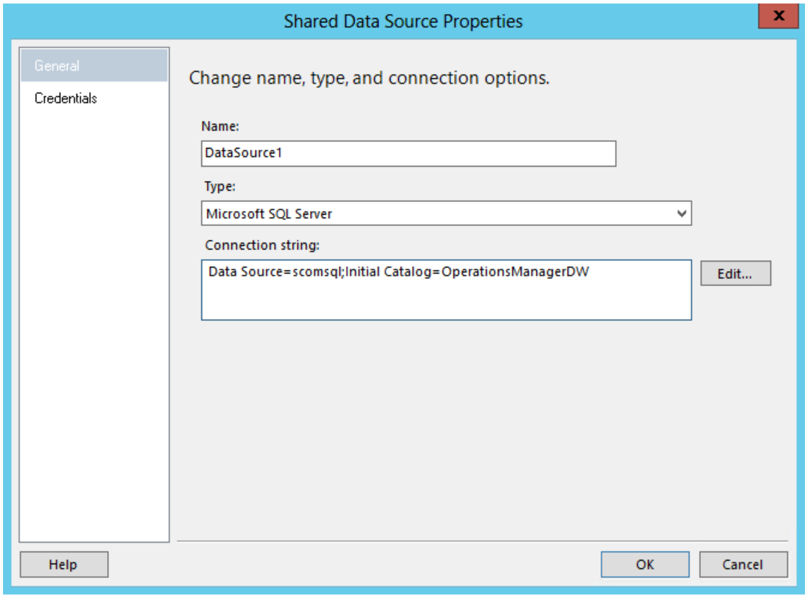
The sample report for this chapter will consist of three sections:

- Current installed agents
- Top CPU users over the last seven days
- Total free space for all logical disks

Just like in the previous chapter, a data source is needed to pull the data from the data warehouse for use in the report.

Adding a data source

To add a new data source using Solution Explorer, right-click Shared Data Sources in the Solution Explorer pane, and then click Add New Data Source. This opens the Shared Data Source Properties dialog box shown in Figure 6-3.



Change name, type, and connection options.

Name: DataSource1

Type: Microsoft SQL Server

Connection string: Data Source=scomsql;Initial Catalog=OperationsManagerDW

FIGURE 6-3 Shared Data Source Properties dialog box

On the General page of the Shared Data Source Properties dialog box, type a name in the Name text box (this should be Data Warehouse Main if you are using the default). Next, click Edit to the right of the Connection String text box, and in the dialog box that opens, type your SQL Server name, for example SERVERNAME\SQLINSTANCENAME. If your environment uses named instances, add the named instance as well.

After you type the SQL Server name in the Server Name text box, you can access the Select Or Enter A Database Name drop-down list. Select your Operations Manager data warehouse database from the drop-down list. By default, the data warehouse database name is OperationsManagerDW, but it may be different in your environment, depending on how it was installed.

After you select the data warehouse, click Test Connection. If the connection is successful, a success message appears. Click OK, and then click OK again in the Connection Properties dialog box to re-access the Shared Data Source Properties dialog box. Click Credentials in the left pane, and then select Windows Authentication. Finally, click OK to return to the design screen where the new data source appears in the Solution Explorer pane.

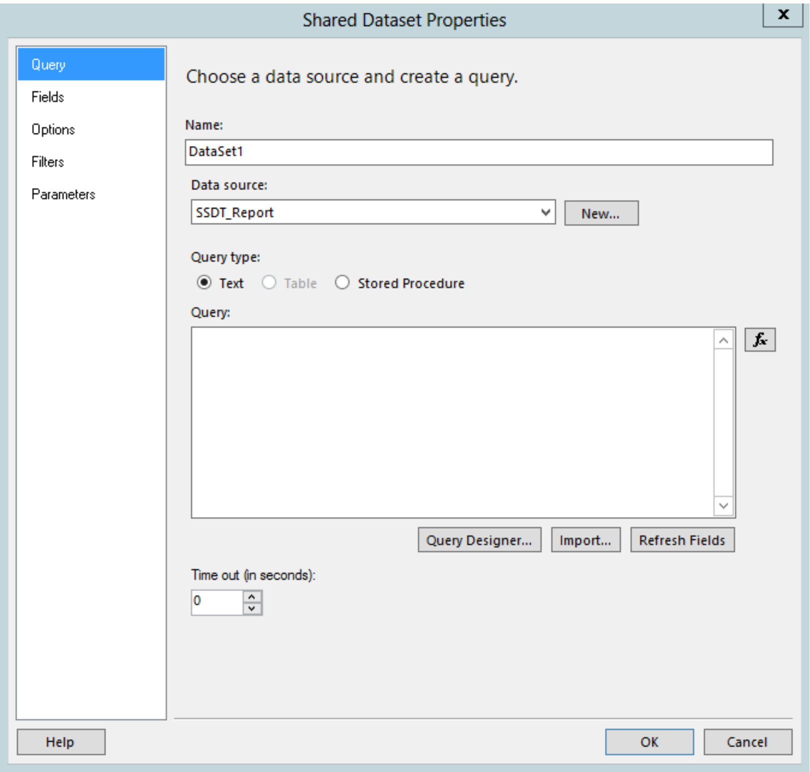
TIP **If you use a shared data source, it must be deployed to SSRS, as in the example in Chapter 5. If the shared data source is not deployed, the reports will fail when they are added to SSRS. One way to avoid this problem is to name the data source Data Warehouse Main in the report so that it references the default data source deployed with Operations Manager Reporting. It is a good idea to use a custom name and deploy the data source manually to avoid corrupting or deleting the default data source when custom reports are removed or modified.**

After the Data Source is added, right-click Reports in the Solution Explorer pane, click Add, click New Item, and then select Report from the list in the box that opens. Give the report a name that will be easy to identify in the Name text box, and click Add. This adds the report to the design screen and to the Report Data pane similar to Report Builder.

NOTE **The preceding procedure reveals a key difference between SSDT and Report Builder. In Report Builder, each report is edited as a standalone object while in SSDT a report project can contain multiple reports. When you add data sources or datasets to the Report Data pane in SSDT, they are added from the project and not from an external source. So if a dataset is added to the project as a shared dataset, it is available for any report inside the project.**

Adding a dataset

To retrieve data using the data source, a dataset first needs to be added. Right-click Shared Datasets in the Solution Explorer pane, and then click Add New Dataset. The Shared Dataset Properties dialog box appears (see Figure 6-4).



Shared Dataset Properties

Choose a data source and create a query.

Name: DataSet1

Data source: SSDT_Report

Query type: ● Text ○ Table ○ Stored Procedure

Query:

Time out (in seconds): 0

FIGURE 6-4 Shared Dataset Properties dialog box

Name the dataset and select the data source created previously. This dataset will be used for the agent version section of the report, so add the query shown in Listing 6-1 to the dataset query box.

LISTING 6-1 Current installed agents query

```
Select Distinct ME.DisplayName,
MPS.FromDateTime,
PST.PropertyValue AS Version From vManagedEntityPropertySet (NoLock) MPS
Inner Join vManagedEntityTypeProperty (NoLock) MEP
On MPS.PropertyGuid = MEP.PropertyGuid
Inner Join vManagedEntity (NoLock) ME
On MPS.ManagedEntityRowId = ME.ManagedEntityRowId
Inner Join vManagedEntityTypeProperty (NoLock) MTP
On MEP.ManagedentityTypeRowId = MTP.ManagedentityTypeRowId
And MTP.PropertySystemName = 'Version'
Inner Join vManagedEntityPropertySet (NoLock) PST
On MTP.PropertyGuid = PST.PropertyGuid
And ME.ManagedentityRowId = PST.ManagedentityRowId
Where MEP.PropertySystemName = 'IsAgent'
And MPS.PropertyValue = '1'
And MPS.ToDateTime Is Null
```

After the query is added, click OK to create the dataset. The dataset should appear in the Solution Explorer pane. The report you are building requires three datasets, so repeat the previous steps to create two more datasets, using the queries in Listing 6-2 and Listing 6-3.

LISTING 6-2 Top CPU users over the last seven days query

```
Create Table #Results(Date VarChar(20),
Value Float,
Server VarChar(100))
Select Convert(VarChar(100),
PD.DateTime, 101) Date ,
MAX(PD.AverageValue) AS Value ,
ME.Path ServerName Into #Temp
From Perf.vPerfDaily PD
Inner Join vManagedEntity ME
On PD.ManagedEntityRowId = ME.ManagedEntityRowId
Inner Join vPerformanceRuleInstance PR
On PD.PerformanceRuleInstanceRowId = PR.PerformanceRuleInstanceRowId
Inner Join vRule R
On PR.RuleRowId = R.RuleRowId
Where R.RuleDefaultName = 'Processor % Processor Time Total Windows Server 2012'
And
PD.DateTime >= DATEADD(Day, -8, GetDate())
Group By Convert(VarChar(100), PD.DateTime, 101), ME.Path
Order By 1,2 Desc
```

```
Declare @OV VarChar(20)
Set @OV = ''
Declare @D VarChar(20),
@V Float, @SName VarChar(100),
@DateTime VarChar(20),
@Value Float,
@Server VarChar(100)
Declare  C1 Cursor For Select Date,
Value, ServerName
From #Temp Order By 1,2 Desc
Open C1
Fetch C1 Into @D,@V,@SName
While @@FETCH_STATUS = 0
Begin
If @OV <> @D
Insert Into #Results(Date, Value, Server)
        Values(@D, @V, @SName)
Set @OV = @D
Fetch C1 Into @D,@V,@SName

End
Close C1
Deallocate C1
Select * From #Results
Drop Table #Temp
Drop Table #Results
```

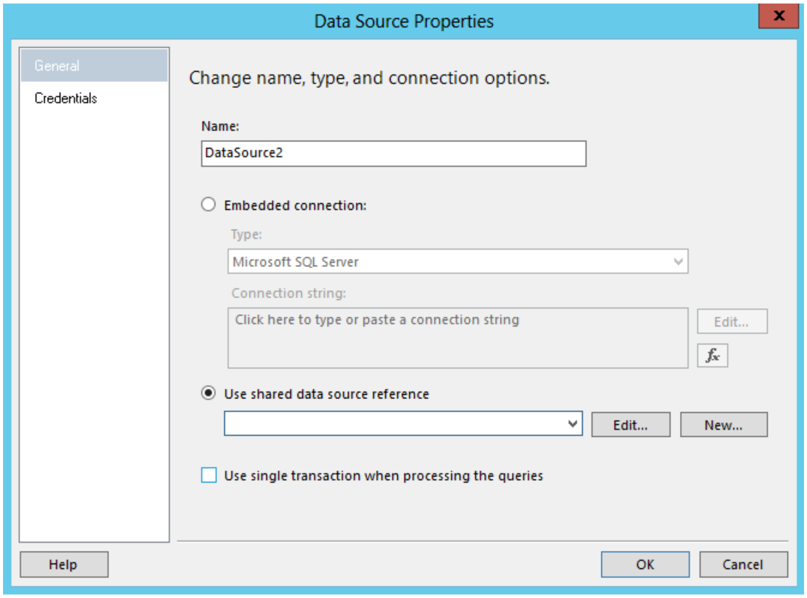
LISTING 6-3 SQL query for logical disk free space

```
SELECT DISTINCT ME.Path
INTO #SNames
FROM Perf.vPerfDaily PD
INNER JOIN vPerformanceRuleInstance PRI
ON PRI.PerformanceRuleInstanceRowId = PD.PerformanceRuleInstanceRowId
INNER JOIN vManagedEntity ME
ON PD.ManagedEntityRowId = ME.ManagedEntityRowId
INNER JOIN vPerformanceRule PR
ON PRI.RuleRowId = PR.RuleRowId
WHERE ObjectName = 'logicaldisk'
SELECT DISTINCT CONVERT(INT, PD.AverageValue) AS FreeSpace,
ME.Path AS ServerName,
R.RuleDefaultName,
ME.Name AS Drive,
PD.DateTime
FROM Perf.vPerfDaily PD
INNER JOIN vManagedEntity ME
```

```
ON PD.ManagedEntityRowId = ME.ManagedEntityRowId
INNER JOIN vPerformanceRuleInstance PR
ON PD.PerformanceRuleInstanceRowId = PR.PerformanceRuleInstanceRowId
INNER JOIN vRule R
ON PR.RuleRowId = R.RuleRowId
WHERE R.RuleDefaultName LIKE '% Logical Disk Free Space Windows Server 2012'
AND PD.DateTime >= DATEADD(DAY, - 2, GetDate())
AND ME.Path
IN (SELECT * FROM #SNames)
DROP TABLE #SNames
```

Adding datasets to a report

When the datasets are added and appear in Solution Explorer, add them to the report by configuring the data source properties. To do this, right-click Data Sources in the Report Data pane, and then click Add Data Source. In the Data Source Properties dialog box, click Use Shared Data Source Reference, and select the data source you created earlier from the drop-down list, as shown in Figure 6-5.



Data Source Properties

Change name, type, and connection options.

Name: DataSource2

○ Embedded connection:

Type: Microsoft SQL Server

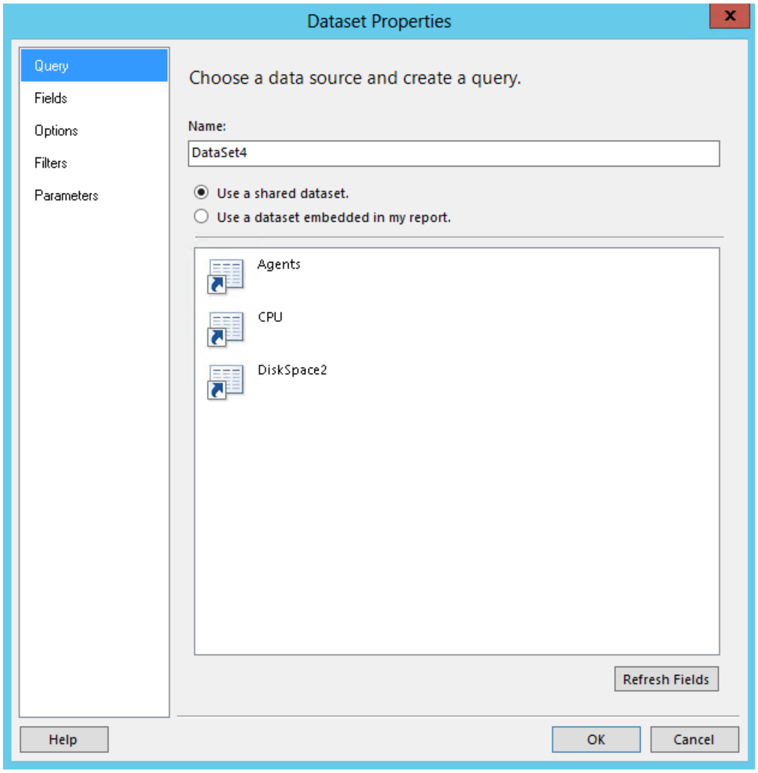
Connection string: Click here to type or paste a connection string

● Use shared data source reference

□ Use single transaction when processing the queries

FIGURE 6-5 Data Source Properties dialog box

To add the datasets to the report, right-click Datasets in the Report Data pane, and then click Add Dataset. In the Dataset Properties dialog box, the three datasets you added to Solution Explorer should be listed. Click the first one you created, name it, and then click OK. It should now appear in the Report Data pane. Repeat these steps to add the other two datasets. Figure 6-6 shows the Dataset Properties dialog box with the three datasets listed.



Choose a data source and create a query.

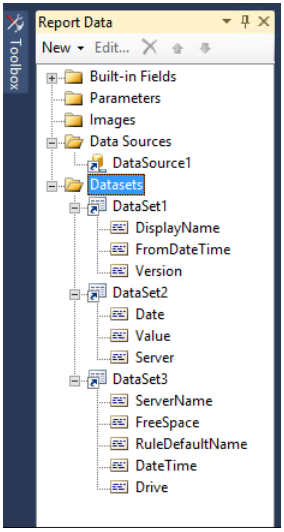
Name:
DataSet4

● Use a shared dataset.
○ Use a dataset embedded in my report.

Agents
CPU
DiskSpace2

FIGURE 6-6 Dataset Properties dialog box

When you are finished, all three datasets should appear in the Report Data pane. Expand each dataset and verify that all fields are included, as shown in Figure 6-7.



- Built-in Fields
- Parameters
- Images
- Data Sources
  - DataSource1
- Datasets
  - DataSet1
    - DisplayName
    - FromDateTime
    - Version
  - DataSet2
    - Date
    - Value
    - Server
  - DataSet3
    - ServerName
    - FreeSpace
    - RuleDefaultName
    - DateTime
    - Drive

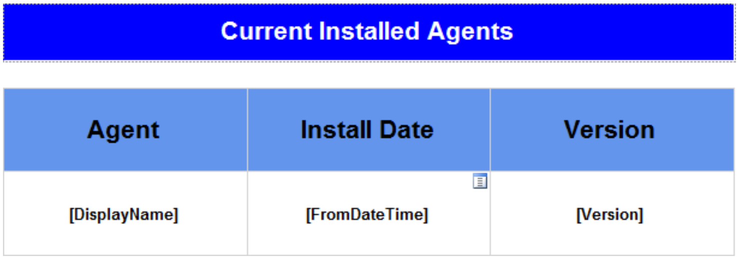
FIGURE 6-7 Report Data pane showing all dataset fields

Designing the report

After all the data is added, you can build your report. Right-click the report in the design screen, click Insert, and then click Text Box. Click inside the added text box and type Current Installed Agents. Right-click below the text box, click Insert, and then click Matrix. The default is two columns, but this report will need three. Right-click the top of the second column, click Insert Column, and then click Outside Group - Left.

When there are three columns, click and hold DisplayName under the Agents dataset and drag it to the first box in the matrix. Repeat this for FromDateTime, dragging it to the bottom box in the second column. Finally, do the same for Version, dragging it to the bottom box in the third column.

Click in the first box in the first column and change the text to Agent. Repeat for the second column, changing the text to Install Date. Change the text in the third column to Version. When finished, the text box and matrix should look like Figure 6-8.



Current Installed Agents

Agent	Install Date	Version
[DisplayName]	[FromDateTime]	[Version]

FIGURE 6-8 Current Installed Agents text box and matrix

Repeat the previous procedure, adding another text box and matrix with three columns. For this text box, change the text to Top CPU Users. For the fields in the matrix, drag Server to the second box in the first column and Value to the second box in the second column. For the final box, right-click in the box, click Insert, and then click Gauge. In the Select Gauge Type dialog box, click the 180 Degrees North gauge, as shown in Figure 6-9, and then click OK.

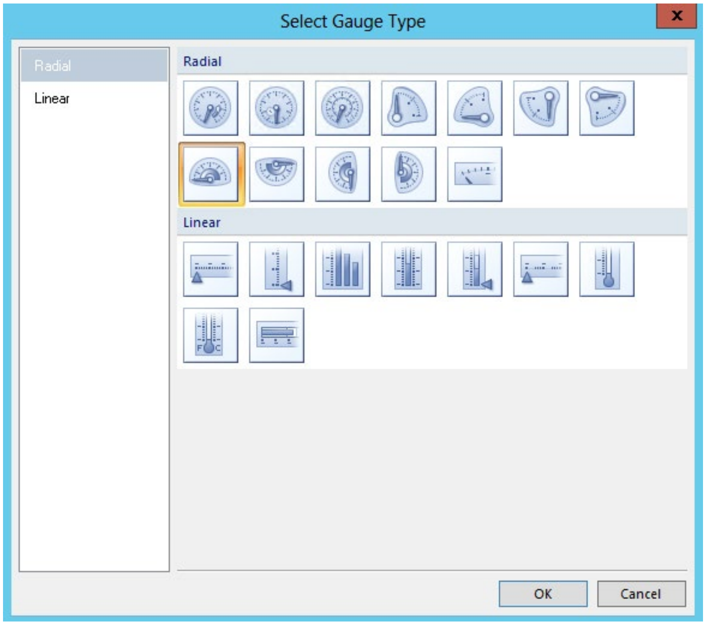
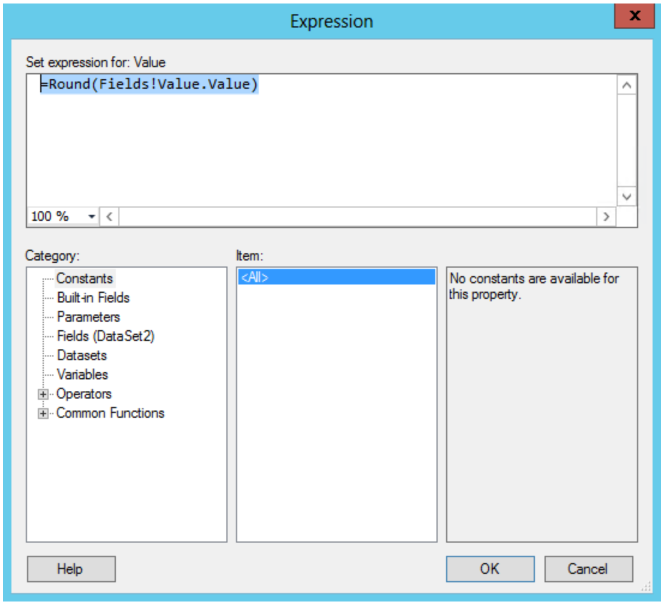


FIGURE 6-9 Select Gauge Type dialog box

After adding the gauge, click it to open the Gauge Data dialog box. Click the down arrow next to RadialPointer1, and click Pointer Properties to open the Pointer Properties dialog box.

In the Pointer Properties dialog box, click the Fx button next to the Value field to open the Radial Pointer Properties dialog box. In the Set Expression For: Value box, type =Round(Fields!Value.Value), as shown in Figure 6-10.

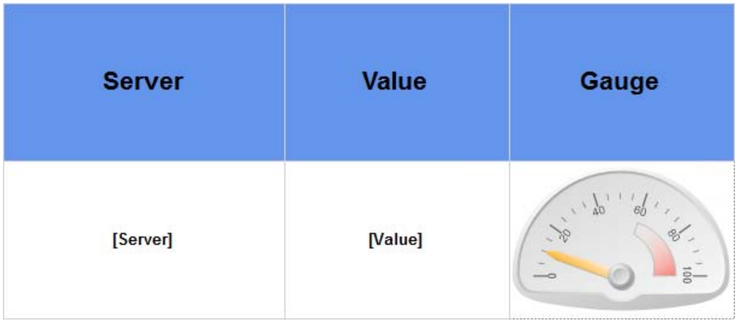


Set expression for: Value

```
=Round(Fields!Value.Value)
```

FIGURE 6-10 Expression dialog box with an expression typed into the Set Expression For Value textbox

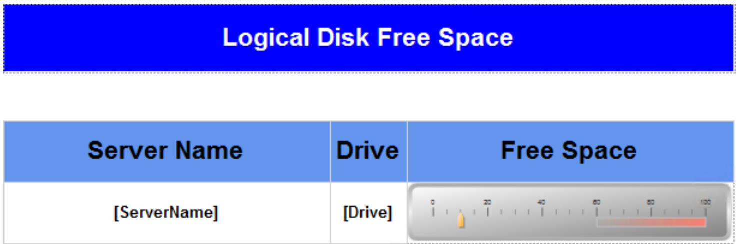
After the expression is added, click OK, and then click OK again in the Radial Pointer Properties dialog box. Next, change the text fields to Server, Value, and Gauge respectively. When everything is finished, the matrix should look like Figure 6-11.



Server	Value	Gauge
[Server]	[Value]	

FIGURE 6-11 Top CPU users text box and matrix

Add another text box and matrix with three columns. Change the text in the text box to Logical Disk Free Space. For the boxes in the matrix, drag ServerName and drive into the first two columns like before. Right-click the bottom box in the third column, click Insert, and then click Gauge. In the Select Gauge Type dialog box, click Linear, click Horizontal, and then click OK. Click the pointer in the gauge to select it, right-click it, and then click Pointer Properties. In the Value field, type [FreeSpace] (including the brackets) and click OK. Change the text values to Server Name, Drive, and Free Space. When everything is finished, the matrix and text box should look like Figure 6-12.



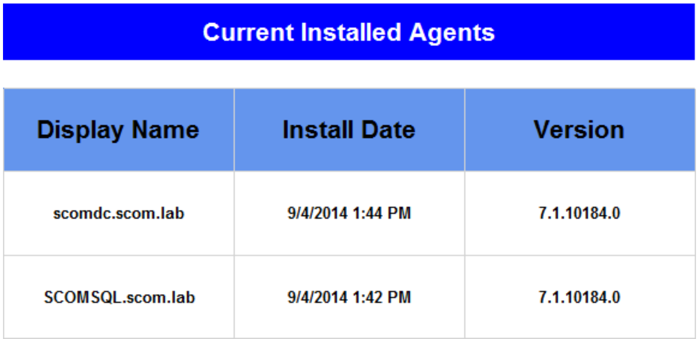
Logical Disk Free Space

Server Name	Drive	Free Space
[ServerName]	[Drive]	

FIGURE 6-12 Logical Disk Free Space text box and matrix

With the sections completed, you can size and move the individual objects to make the report easier to read. You can also alter the colors to improve the appearance of the report. To change the color of any box, right-click it and select (Object Name) Properties for the object whose properties you want to modify, for example Tablix Properties or Text Box Properties. For example, to change the color of the Current Installed Agents text box, click to highlight it, right-click inside the box, and then click Text Box Properties. When the properties dialog box opens, click Fill, and select a new fill color. To change the font, click Font and select a new font from the list.

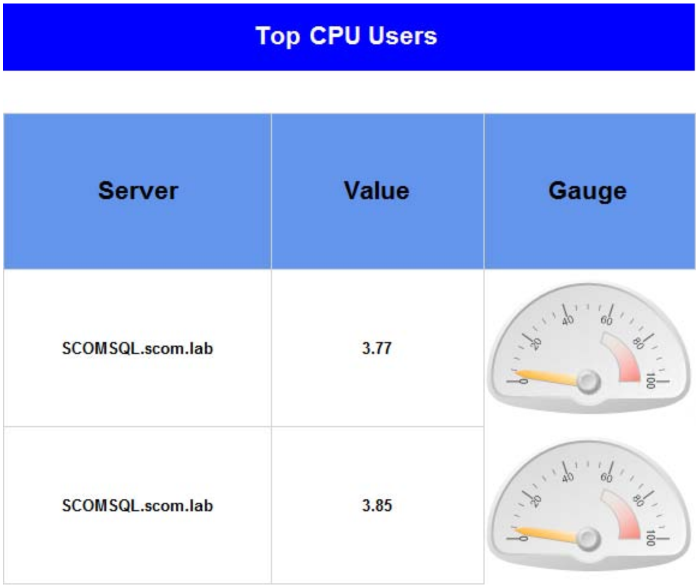
Figures 6-13, 6-14, and 6-15 show screen captures of the completed sections.



Current Installed Agents

Display Name	Install Date	Version
scomdc.scom.lab	9/4/2014 1:44 PM	7.1.10184.0
SCOMSQL.scom.lab	9/4/2014 1:42 PM	7.1.10184.0

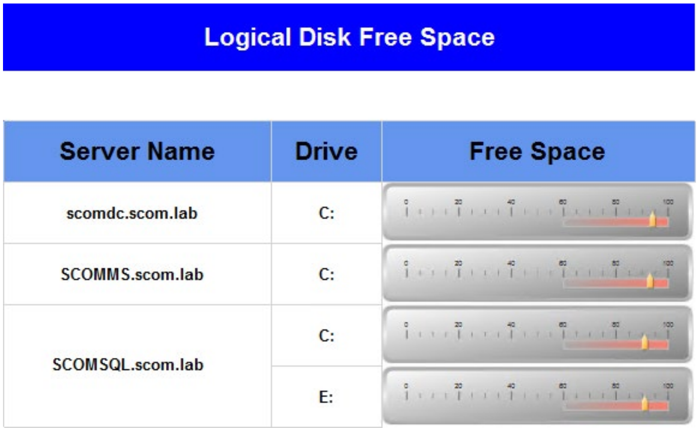
FIGURE 6-13 Current installed agents



Top CPU Users

Server	Value	Gauge
SCOMSQL.scom.lab	3.77	
SCOMSQL.scom.lab	3.85	

FIGURE 6-14 Top CPU users over the last seven days



Logical Disk Free Space

Server Name	Drive	Free Space
scomdc.scom.lab	C:	
SCOMMS.scom.lab	C:	
SCOMSQL.scom.lab	C:	
	E:	

FIGURE 6-15 Logical disk free space

The final step is to localize the report. The steps for doing this are covered in Chapter 7, "Building management packs for reporting," in the section titled "Dataset," and Listing 7-8 shows an example of the required XML.

CHAPTER 7

Building management packs for reporting

This chapter is about leveraging a management pack to store the information you create in your reports. You will learn about the different management pack elements used for reporting, how to create a management pack to store your reports, and how and when you may need to bundle all of the parts together. Storing your reports and their dependencies within a management pack is a good practice because it makes your reports flexible and easy to import into other environments. To get the most from this chapter, you should have a general knowledge of the management pack structure. The Management Pack Authoring Guide on TechNet is a good resource to refresh your knowledge (see *http://technet.microsoft.com/en-us/library/ee957010.aspx*).

Some benefits of leveraging a management pack include:

- The out-of-box structure provided by the management pack schema
- Support for localization of report labels and prompts
- The ability to add, upgrade, and remove reports and their artifacts when you import, upgrade, or remove the management pack from the management group

Understanding reporting sections within a management pack

Management packs are structured XML data used to configure System Center Operations Manager. While not specifically discussed in this book, the schema files that determine what can be in a management pack can be found in the Operations Manager Management Server installation directory. All of the schema files have the .xsd extension. These files can help you understand the required structure of a management pack. The schema leveraged when creating a management pack largely depends on which version of Operations Manager you are using. Management packs created for Operations Manager 2007 R2 must leverage the v1.1 schema, while management packs created for Operations Manager 2012 and newer versions leverage the v2.0 schema. Management packs created with the v1.1 schema can be imported in an Operations Manager 2012 environment, but they can’t take advantage of the benefits of the v2.0 schema.

Reporting follows the schema just like other aspects of Operations Manager. Reporting depends on three major sections of the management pack schema:

- Presentation
- Reporting
- Language Packs

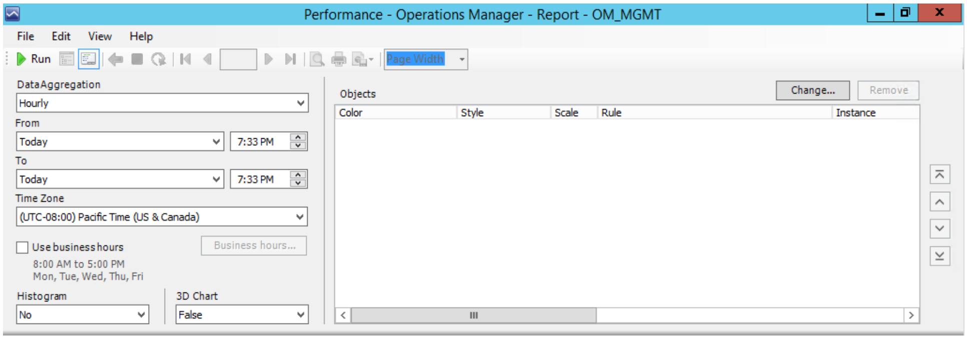
Presentation

The Presentation section contains the StringResources element. String resources localize the labels for report parameters during report generation and display. The example in Listing 7-1 shows multiple string resources within the Presentation section. Each of these string resource IDs will be referenced by an element in the Report Parameters section, which is explained a little later in this chapter. The string resource IDs will also be referenced directly from the Display Strings section of your management pack to provide the display names and the localization of the parameters.

LISTING 7-1 Presentation tag with the StringResources element

```
<Presentation>
   <StringResources>
     <StringResource ID="Example.ParameterPrompt.Prompt1" />
     <StringResource ID="Example.ParameterPrompt.Prompt2" />
     <StringResource ID="Example.ParameterPrompt.Prompt3" />
   </StringResources>
</Presentation>
```

Figure 7-1 is an example of the Report Parameters section of a report within the Operations Manager console. The elements shown, such as DataAggregation, From, To, Objects, and all the rest of the labels, are localized by referencing the string resources defined in the Presentation section.



Performance - Operations Manager - Report - OM_MGMT

DataAggregation: Hourly

From: Today 7:33 PM

To: Today 7:33 PM

Time Zone: (UTC-08:00) Pacific Time (US & Canada)

☐ Use business hours

8:00 AM to 5:00 PM
Mon, Tue, Wed, Thu, Fri

Histogram: No

3D Chart: False

Objects

Color	Style	Scale	Rule	Instance

FIGURE 7-1 Report labels localized by referencing a StringResources element

You could use a table to track the labels of related elements that appear in different sections of the management pack. For example, in Table 7-1, the first column shows the actual name of the report parameter, the middle column shows the string resource ID, and the last column shows the corresponding display string.

TABLE 7-1 Using a table to document related management pack elements

REPORT PARAMETER NAME	STRING RESOURCE	DISPLAY STRING
FromDate	Example.ParameterPrompt.FromDate	From

Reporting

The Reporting section contains the majority of the elements that are used for storing a report or reports within a management pack. Whereas the Presentation and Language Pack sections are helper sections that allow you to extend reporting to support different languages, the Reporting section includes all of the main reporting elements. The following is a list of some of the elements within the Reporting section:

- DataWarehouseScripts
- ReportResources
- Reports
- LinkedReports
- ReportParameterControls

> **NOTE** **There are other elements in the Reporting section that are not discussed in this book. To get more information on these sections and many others, see *http://msdn.microsoft.com/en-us/library/ee533489.aspx*.**

The Data Warehouse Scripts section contains the scripts that install, upgrade, or uninstall each report in your management pack. Depending on the tool you use to create the management pack, the DataWarehouseScripts element can differ. In the v1.1 schema, you paste the install script directly into the Install element. This can be problematic because SQL scripts sometimes include angle brackets (< >), which causes issues with the XML. To deal with this in the v1.1 schema, you can put the SQL query within a CDATA Section <![CDATA[script goes here]]>. The v2.0 schema addresses this issue by referencing a resource instead. However, referencing an outside file (i.e., a SQL query file) greatly changes the management pack and requires that you create a management pack bundle (.mpb) file. Listing 7-2 shows an example of the DataWarehouseScripts element in the v1.1 schema while Listing 7-3 shows the same example leveraging the v2.0 schema.

LISTING 7-2 DataWarehouseScripts element in v1.1 schema

```
<DataWarehouseScripts>
   <DataWarehouseScript ID="Example.Report.Script" Accessibility="Public">
     <Install>
     <!-- Stored Procedure goes here -->
     </Install>
     <Uninstall>
     <!-- Stored Procedure goes here -->
     </Uninstall>
     <Upgrade>
     <!-- Stored Procedure goes here -->
     </Upgrade>
   </DataWarehouseScript>
</DataWarehouseScripts>
```

LISTING 7-3 DataWarehouseScripts element in v2.0 schema

```
<DataWarehouseScripts>
   <DataWarehouseScript ID="Example.Report.Script" Accessibility="Public">
     <InstallScript>Contoso.Example.Report.InstallScript</InstallScript>
     <UninstallScript>Contoso.Example.Report.UninstallScript</UninstallScript>
     <UpgradeScript>Contoso.Example.Report.InstallScript</UpgradeScript>
   </DataWarehouseScript>
</DataWarehouseScripts>
```

Report Resources is another configurable setting for reporting used to add external resources to the report, for example an image. This element also differs between the v1.1 and v2.0 schemas. In the v1.1 schema, the Report Resources setting is defined in the Reporting element, where you must define the MIMEType for the resource. In Listing 7-4, for example, the resource is an image, so the MIMEType is set as image/png. Next you must define the data for the image by listing the base64 string of the image. This is the base64 string equivalent of the image itself, which can be obtained via Windows PowerShell using the example in Listing 7-5.

LISTING 7-4 ReportResources element in the v1.1 schema

```
<ReportResources>
   <ReportResource ID="Example.Image.png" Accessibility="Public" MIMEType="image/png">
     <ResourceData><!-- Content goes here --></ResourceData>
   </ReportResource>
</ReportResources>
```

LISTING 7-5 Converting an image to a base64 string with Windows PowerShell

```
$path = "C:\Users\gwallace\Desktop\ISpl9za2tm75c41000000000.jpg"
[convert]::ToBase64String((get-content $path -encoding byte))
```

In the v2.0 schema, the Report Resources setting is defined in the Resources element at the bottom of the management pack. This makes adding report resources, such as images, report definition files (RDL), and other items, easier since they will be bundled with the management pack. There is no need to paste the entire script in the management pack or to convert your images to base64 encoded strings; Operations Manager will handle everything for you when you bundle it together. An example of the v2.0 schema format is shown in Listing 7-6.

LISTING 7-6 Resources element in the v2.0 schema

```
<Resources>
   <ReportResource ID="Contoso.Example.Report1.ReportRdl" Accessibility="Public"
FileName="Contoso.Example.Report1.rdl" HasNullStream="false"
MIMEType="application/octet-stream" />
</Resources>
```

The Report Strings section localizes values for the report labels, similar to how the string resources localize the parameters of the report. Listing 7-7 shows an example of the ReportStrings element. Each report string is simply an ID that is targeted as the ElementID in the display string, which localizes the report labels. The report references the report string, which in turn references a display string.

LISTING 7-7 ReportStrings element

```
<ReportStrings>
   <ReportString ID="RS.ReportName" />
   <ReportString ID="RS.ReportDescriptionLabel" />
   <ReportString ID="RS.ReportDescription" />
</ReportStrings>
```

For localization of the report to work, you need to add a new dataset, along with two additional report parameters and some custom code to the .rdl file.

Dataset

The dataset you create will obtain the report strings from the management group and add them to the .rdl file. Listing 7-8 shows an example of a dataset. The dataset will be used for one of the report parameters, which you will define later. This dataset targets the ReportDisplayStringGet stored procedure within the OperationsManagerDW database. This stored procedure requires five parameters. The first required parameter is ManagementGroupGuid, which will be provided via code defined later in this chapter. Next, you must hard code the management pack ID and the report ID (these values will never change). Another parameter will get the current language code for the report user. The final required parameter is the scope, which can be used to limit the results. In the example shown in Listing 7-8, the scope value is set to percent (%), which is a wildcard for all values.

LISTING 7-8 LT_Report_Query dataset

```
<DataSet Name="LT_Report_Query">
   <Query>
     <DataSourceName>DataWarehouseMain</DataSourceName>
     <QueryParameters>
        <QueryParameter Name="@ManagementGroupGuid">
          <Value>=Code.GetCallingManagementGroupId()</Value>
        </QueryParameter>
        <QueryParameter Name="@ManagementPackSystemName">
          <Value>Contoso.Example.Report.Library</Value>
        </QueryParameter>
        <QueryParameter Name="@ReportSystemName">
          <Value>Contoso.Example.Report.Library.ExampleNewReport</Value>
        </QueryParameter>
        <QueryParameter Name="@LanguageCode">
          <Value>=Code.GetReportLocLanguageCode()</Value>
        </QueryParameter>
        <QueryParameter Name="@Scope">
          <Value>%</Value>
        </QueryParameter>
     </QueryParameters>
     <CommandType>StoredProcedure</CommandType>
     <CommandText>ReportDisplayStringGet</CommandText>
   </Query>
   <Fields>
     <Field Name="ValueXml">
        <DataField>XML_F52E2B61-18A1-11d1-B105-00805F49916B</DataField>
        <rd:TypeName>System.String</rd:TypeName>
     </Field>
     <Field Name="ValueName">
        <Value>="LT"</Value>
     </Field>
   </Fields>
</DataSet>
```

Report parameters

After you add the dataset, you must add the report parameters to the .rdl file, as shown in Listing 7-9. These are added within the ReportParameter element. The first parameter stores the values returned by the LT_Report_Query dataset defined in Listing 7-8. This is where the code will obtain all of the display string values. The next parameter indicates the language code needed for the person running the report; this parameter calls the code directly.

LISTING 7-9 Report parameters

```
<ReportParameter Name="LT_Report">
   <DataType>String</DataType>
   <DefaultValue>
     <DataSetReference>
        <DataSetName>LT_Report_Query</DataSetName>
        <ValueField>ValueXml</ValueField>
     </DataSetReference>
   </DefaultValue>
   <Hidden>true</Hidden>
</ReportParameter>
<ReportParameter Name="LangCode">
   <DataType>String</DataType>
   <DefaultValue>
     <Values>
        <Value>=Code.GetReportLocLanguageCode()</Value>
     </Values>
   </DefaultValue>
   <Hidden>true</Hidden>
</ReportParameter>
```

Custom code

The last section to add to the RDL file to ensure localized content is the code itself. This code must be added within the Code element. Listing 7-10 shows an example of the code contents. This code block contains the functions and variables needed to support the parameters and dataset described in the previous sections.

LISTING 7-10 Code required for report localization

```
<Code>
Const LocTableStringQuery As String = "R/S[@N='{0}']/text()"
Dim LocTables As System.Collections.Generic.Dictionary(Of String,
   Microsoft.EnterpriseManagement.Reporting.XmlStringTable)
Dim ReportTimeZone As Microsoft.EnterpriseManagement.Reporting.TimeZoneCoreInformation
Dim ReportStartDate As DateTime
Dim ReportEndDate As DateTime
Dim ReportTime As
Microsoft.EnterpriseManagement.Reporting.ParameterProcessor.RelativeTime
Dim ReportCulture As System.Globalization.CultureInfo
Dim ParameterProcessor As Microsoft.EnterpriseManagement.Reporting.ParameterProcessor
Dim TargetList As String

Public Function GetLocTable(Name As String) As
Microsoft.EnterpriseManagement.Reporting.XmlStringTable
Dim LocTable As Microsoft.EnterpriseManagement.Reporting.XmlStringTable
```

```
If Not LocTables.TryGetValue(Name, LocTable) Then
LocTable = New
Microsoft.EnterpriseManagement.Reporting.XmlStringTable(LocTableStringQuery,
   Report.Parameters(Name).Value)
LocTables.Add(Name, LocTable)
End If
Return LocTable
End Function

Public Function GetReportLocLanguageCode() As String
Return ReportCulture.ThreeLetterWindowsLanguageName
End Function

Public Function GetCallingManagementGroupId() As String
Return Microsoft.EnterpriseManagement.Reporting.ReportingConfiguration.ManagementGroupId
End Function

Protected Overrides Sub OnInit()
LocTables = new System.Collections.Generic.Dictionary(Of String,
Microsoft.EnterpriseManagement.Reporting.XmlStringTable)()
ReportTimeZone =Nothing
ReportStartDate = DateTime.MinValue
ReportEndDate = DateTime.MinValue
ReportTime = Nothing
ReportCulture = System.Globalization.CultureInfo.GetCultureInfo(Report.User("Language"))
ParameterProcessor = New
Microsoft.EnterpriseManagement.Reporting.ParameterProcessor(ReportCulture)
TargetList =Nothing
End Sub
</Code>
<CodeModules>
   <CodeModule>Microsoft.EnterpriseManagement.Reporting.Code, Version=6.0.0.0,
      Culture=neutral</CodeModule>
</CodeModules>
```

To localize an element within the .rdl file, enter the code shown in Listing 7-11 for the expression of the text box or element.

LISTING 7-11 Report parameter

```
=Code.GetLocTable("LT_Report").GetString("ReportStringExample1",
   "[ExampleDefaultTexttoDisplay]")
```

Language packs

Language packs are powerful tools for Operations Manager. They allow you to create reports, new rules, monitors, or almost any extension for Operations Manager in an end user's native language. Each language is represented by a Language Pack section containing multiple display strings. Listing 7-12 shows one example.

LISTING 7-12 Language Pack section

```
<LanguagePack ID="ENU" IsDefault="true">
   <DisplayStrings>
     <DisplayString ElementID="Example.ParameterPrompt.Prompt1"
        <Name>Report Prompt 1</Name>
     </DisplayString>

     <DisplayString ElementID="Example.ParameterPrompt.Prompt2"
        <Name>Report Prompt 2</Name>
     </DisplayString>

   </DisplayStrings>
</LanguagePack>
```

Creating the management pack

To create a management pack, you need the following:

- Scripts to create stored procedures (for install and uninstall)
- Report definition file (RDL)

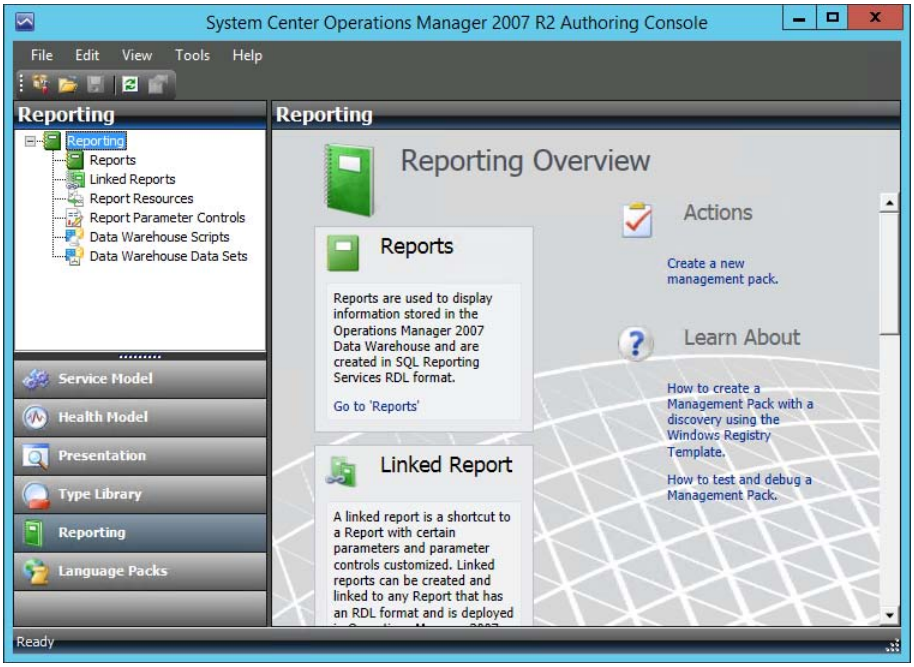
Many management pack authoring tools are available, as described in Chapter 4, but just a few tools can be used to simplify adding a report to a management pack. Currently, two tools support reports: System Center Operations Manager 2007 R2 Authoring Console and System Center 2012 Visual Studio Authoring Extensions. Operations Manager 2007 R2 Authoring Console is the easiest tool for reporting. However, although it is still supported, there will be no more updates to the product, and you could encounter compatibility issues since management packs created with this tool leverage the v1.1 schema whereas Operations Manager 2012 R2 currently uses version 2.0.

COMPANION CONTENT **The management pack example described in this chapter is provided in the companion files. Management packs following both the v1.1 schema and the v2.0 schema are included.**

Operations Manager 2007 R2 Authoring Console

Operations Manager 2007 R2 Authoring Console was created originally for Operations Manager 2007 R2, but it remains a great tool for creating management packs. It is a good blend of MP Author by Silect and System Center 2012 Visual Studio Authoring Extensions. It offers more robust functionality than MP Author in that you can create deeper rules and leverage certain module capabilities of Operations Manager. But it is not a "Click, Next" type of tool; you need to know what you are doing. However, as compared to Visual Studio Authoring Extensions, this tool provides a good framework to work with and a much more guided experience, while still offering deep configuration. The one option unique to the Authoring Console is easy report creation. MP Author does not currently offer any reporting capability, and System Center 2012 Visual Studio Authoring Extensions only configures reports in empty management pack fragments, which means you must write the XML yourself with only Intellisense to help you.

Figure 7-2 shows the Authoring Console, which closely resembles the structure of the Operations Manager console. In the lower left corner are the workspaces for the sections of the tool, including the Presentation, Reporting, and Language Packs sections. The figure shows how the Authoring Console follows a structure similar to the schema.



Reports are used to display information stored in the Operations Manager 2007 Data Warehouse and are created in SQL Reporting Services RDL format.

Go to 'Reports'

A linked report is a shortcut to a Report with certain parameters and parameter controls customized. Linked reports can be created and linked to any Report that has an RDL format and is deployed

Create a new management pack.

How to create a Management Pack with a discovery using the Windows Registry Template.

How to test and debug a Management Pack.

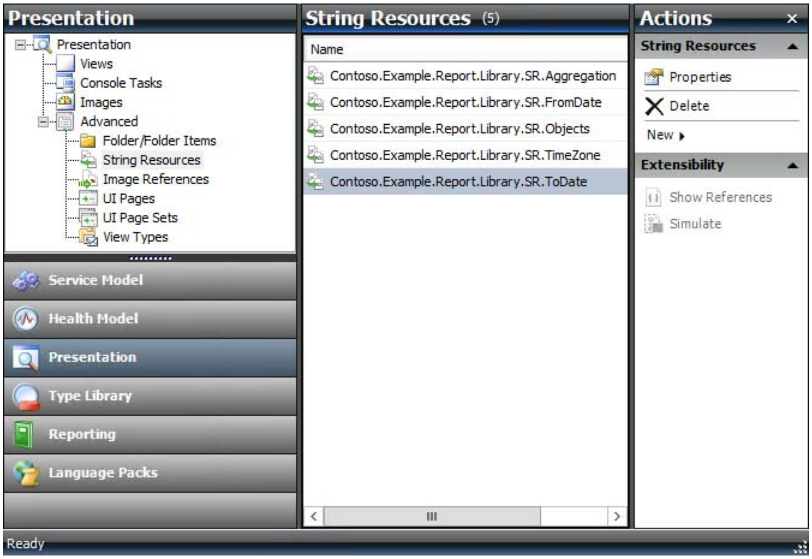
FIGURE 7-2 Operations Manager 2007 R2 Authoring Console

MORE INFO Details for using the Operations Manager 2007 R2 Authoring Console are documented on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/ee957010.aspx*.

To create a new management pack from within the Authoring Console, click File and then New from the menu bar, enter a descriptive internal name for the file name in the Management Pack Identity text box, choose the Empty Management Pack template, and then click Create. Next, you can begin configuring the different sections of the management pack.

Configuring the string resources

You add and view the string resources in the Presentation workspace. As described before, the string resources are used within the report parameter controls to localize the report parameter labels. Figure 7-3 shows the Presentation workspace with several string resources listed.



Contoso.Example.Report.Library.SR.Aggregation
Contoso.Example.Report.Library.SR.FromDate
Contoso.Example.Report.Library.SR.Objects
Contoso.Example.Report.Library.SR.TimeZone
Contoso.Example.Report.Library.SR.ToDate

FIGURE 7-3 String resources in the Presentation workspace

To add a new string resource, click New under Actions and then click New String Resource. In the Choose a unique identifier window, give the string resource a descriptive name and click OK. On the next page, give the string resource a display name and description. Repeat this process for each parameter defined in your report.

Adding the data warehouse scripts

The next step is to add the data warehouse scripts in the Reporting workspace. As described earlier, each script element is split into three parts: an install script, an uninstall script, and an upgrade script. When you add new data warehouse scripts in the Authoring Console, you must paste each script into its respective location; many times the upgrade and install scripts will be the same script. Figure 7-4 shows the Reporting workspace with one data warehouse script added.

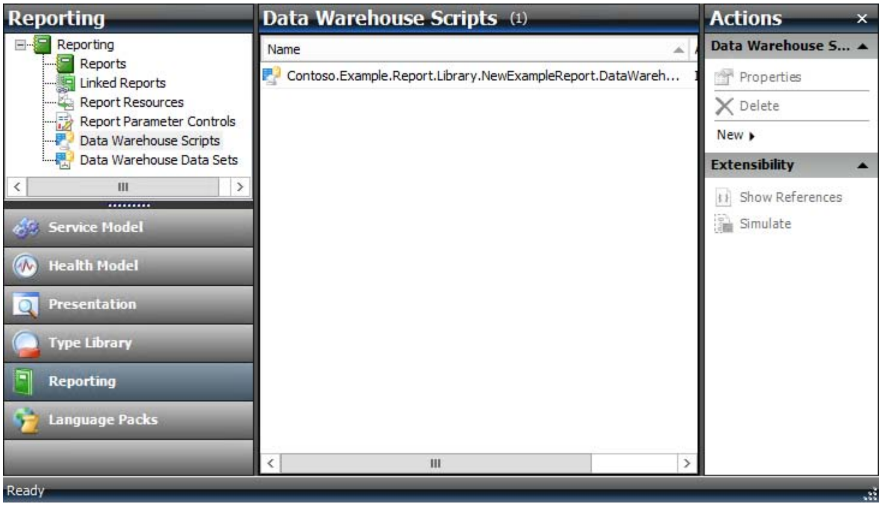
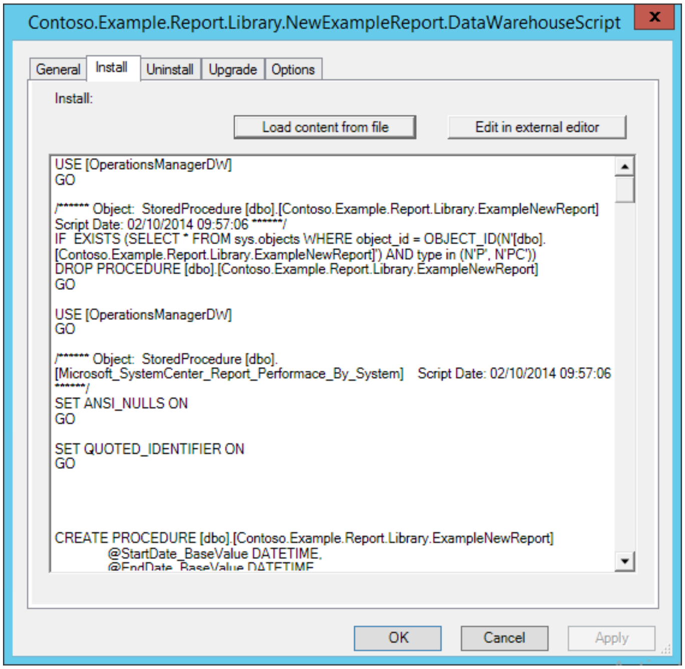


FIGURE 7-4 A data warehouse script in the Reporting workspace

To add a data warehouse script, click in the Reporting workspace and click the Data Warehouse Scripts element. Then click New in the Actions pane and click New Data Warehouse Script. In the Choose The Identifier window, give the script a unique internal name and click OK. In the script window that opens, give the script a display name and description. To set up the SQL scripts, click Install and either paste the install script directly in the text area or identify a SQL file to load in the window. Figure 7-5 shows an example of a script pasted in the text area.

NOTE **When inserting data with angle brackets (< >) into XML elements, surround it with the CDATA tag to ensure the data is imported correctly:**

```
<![CDATA[ Your data goes here ]]>
```



```
USE [OperationsManagerDW]
GO

/****** Object:  StoredProcedure [dbo].[Contoso.Example.Report.Library.ExampleNewReport]
Script Date: 02/10/2014 09:57:06 ******/
IF  EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].
[Contoso.Example.Report.Library.ExampleNewReport]') AND type in (N'P', N'PC'))
DROP PROCEDURE [dbo].[Contoso.Example.Report.Library.ExampleNewReport]
GO

USE [OperationsManagerDW]
GO

/****** Object:  StoredProcedure [dbo].
[Microsoft_SystemCenter_Report_Performace_By_System]    Script Date: 02/10/2014 09:57:06
******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO



CREATE PROCEDURE [dbo].[Contoso.Example.Report.Library.ExampleNewReport]
        @StartDate_BaseValue DATETIME,
        @EndDate_BaseValue DATETIME
```

FIGURE 7-5 A SQL script for creating a stored procedure

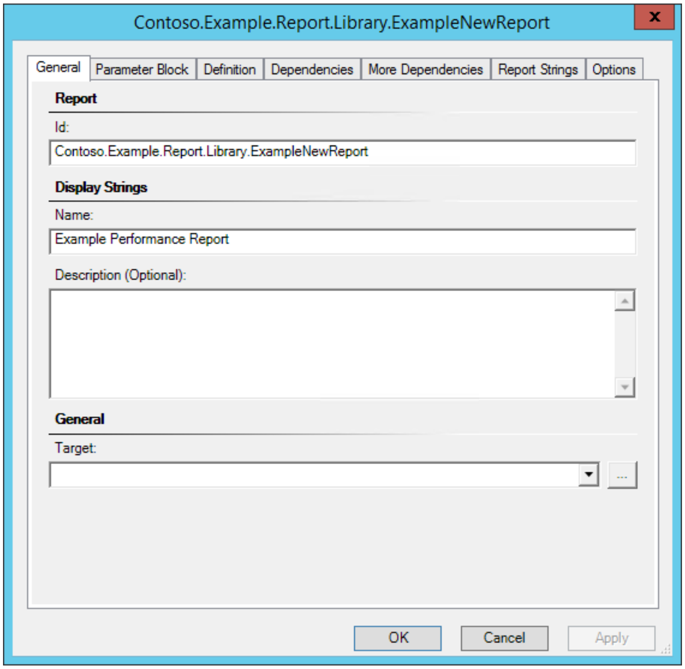
Complete the same procedure for the uninstall and upgrade scripts. At the end of the install and upgrade scripts, it is critical that you add the code from Listing 7-13. This ensures that the reader account has permission to generate the report. If this code is not added, the user will get an insufficient permissions error when attempting to run the report.

LISTING 7-13 Code for run permission for stored procedures

```
GRANT EXECUTE ON [dbo].[Example_Performance_Report] TO OpsMgrReader
GO
```

Configuring the report

To configure the report from the Authoring Console, click the Reporting workspace, click the Reports element, click New, and then click New Report. In the Choose A Unique Identifier window give the report a unique id and click OK. On the General tab of the Report properties window type, a display name in the Name text box. There are seven tabs of configuration settings as shown in Figure 7-6.



Report

Id: Contoso.Example.Report.Library.ExampleNewReport

Display Strings

Name: Example Performance Report

Description (Optional):

General

Target:

FIGURE 7-6 Report properties window in the Authoring Console

To configure the settings on the Parameter Block tab, imagine the ParameterBlock as a grid. Define the number of columns in the ParameterBlock code to format the parameters. Within the ParameterBlock code are the controls that will be displayed on the screen. Listing 7-14 shows an example of the ParameterBlock section. On each control within ParameterBlock you can define the number of columns and rows within the grid with the columnspan and rowspan parameters. Define the parameter within each control. The ReportParameter element ties back to the Report Parameters section in the RDL. The Prompt element ties back to the String Resources section defined previously; this is where you insert the ID of the corresponding string resources.

LISTING 7-14 ParameterBlock code example

```
<ParameterBlock columns="6"
xmlns="http://schemas.microsoft.com/mom/reporting/2007/ReportParameterSettings">
<Controls>
<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.
   RelativeDateTimePicker" rowSpan="2" columnSpan="2">
<ReportParameters>
<ReportParameter name="TimeZone" binding="TimeZone">
<Prompt>Example.ParameterPrompt.Prompt1</Prompt>
</ReportParameter>
</ReportParameters>
</Control>
```

```
...
</Controls>
</ParameterBlock>
```

MORE INFO A list of common controls can be found on TechNet at *http://technet.microsoft.com/en-us/library/dd362610.aspx#sectionSection1*. This list refers to Operations Manager 2007 R2, but the controls have not changed.

Report Definition is the next section to configure. This section houses the code from the .rdl file that you learned how to create earlier in the book. An .rdl file follows an xml schema. Copy the code from the rdl file and paste it into the text box on the Definition tab. After pasting the code from the .rdl file, remove the first xml element at the beginning of the report. Also, replace the DataSource that was configured when using the report with the DataSource element as shown in Listing 7-15.

LISTING 7-15 ParameterBlock code example

```
<DataSource Name="DataWarehouseMain">
   <DataSourceReference>Data Warehouse Main</DataSourceReference>
   <rd:SecurityType>None</rd:SecurityType>
   <rd:DataSourceID>e4235c51-407f-4065-8519-a1e57374bc45</rd:DataSourceID>
</DataSource>
```

You configure the report dependencies on the Dependencies and More Dependencies tabs. Report dependencies define what must be present in the Operations Manager management group for the management pack to import successfully. Configuring report dependencies is not required to store your report in a management pack, but it is a good practice to follow. A dependency is required only when you create a new report parameter that is saved in a different management pack or when you use a custom one. Two examples of dependencies are data warehouse scripts and report parameter controls. For instance, a good practice is to add the data warehouse script that you define for your report as a dependency, although if the script is part of the same management pack, adding it is not necessary. Also, you can define the report parameter controls that your report requires. Again, this is not necessarily required unless you have created a custom control in another management pack.

TIP During import, the management pack will look for the dependencies and if it cannot find them, you will get an error.

Define the report strings next on the Report Strings tab. Report strings are used to link elements within a report to display strings in your management pack. As described previously, this requires adding custom code to your report definition. In the Authoring Console on the Report Strings tab of the Report properties window, click Create, give the report string a unique internal name, and click OK. By default, the Report window will reappear. You must edit the string you just created and give it a display name to properly localize the report string.

Visual Studio Authoring Extensions

Previously in this chapter, you read about using the Operations Manager 2007 R2 Authoring Console to create a management pack to house your reports. As described, the Authoring Console provides an intuitive interface with specific text boxes in which to paste data or options to select items from within the management pack. You can also leverage Visual Studio Authoring Extensions for System Center to create your management pack.

NOTE Visual Studio Authoring Extensions does not provide a management pack fragment for reports. However, included in the companion files for this book is a management pack fragment (ReportFragment.mpx) to use for management pack configuration as well as the Visual Studio 2013 project contained in the Contoso.Example.Report.Library.zip file that is referenced in this chapter. Since Visual Studio Authoring Extensions is meant for Operations Manager 2012, the v2.0 schema is used when building a reporting management pack with this tool.

First add the Authoring Extensions to Visual Studio. The Authoring Extensions can be found at the Download Center (see *http://www.microsoft.com/en-us/download/details.aspx?id=30169*). After they are installed, when you create a new project in Visual Studio, you will see the management pack template options. Select the Operations Manager 2012 management pack template that matches your environment, give it an internal name, and click OK. An empty management pack with no elements and the basic management pack references is created (see Figure 7-7).

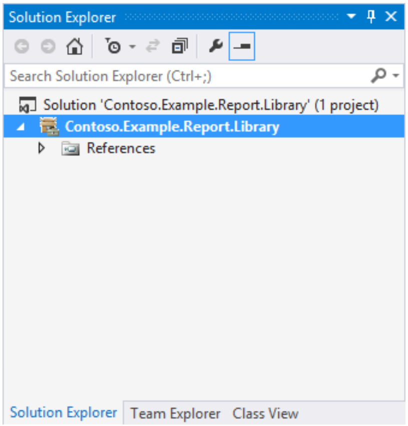
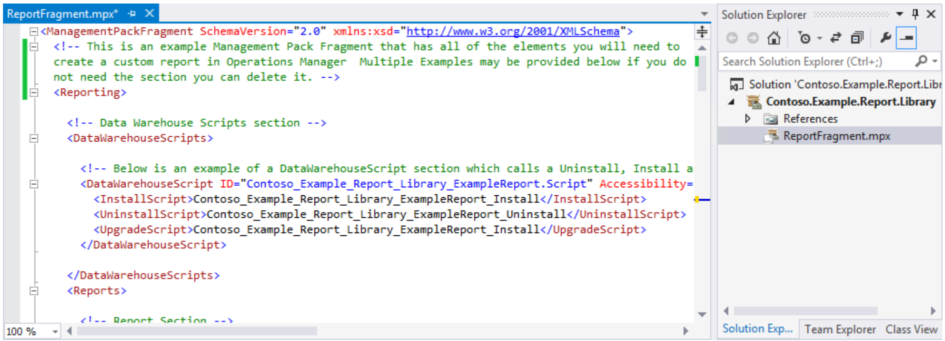

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Contoso.Example.Report.Library' (1 project)
Contoso.Example.Report.Library
References
Solution Explorer
Team Explorer
Class View

FIGURE 7-7 Empty management pack

After the empty management pack has been created, you can configure the management pack properties, such as the name, whether or not it will be sealed, and deployment options. An article describing this process can be found on TechNet at *http://social.technet.microsoft.com/wiki/contents/articles/5236.visual-studio-authoring-extensions-for-system-center-2012-operations-manager.aspx#Setting_Management_Pack_Properties*.

Next, you must add a new management pack fragment to assist in creating the management pack. To do so, right-click the project, click Add, and then click Existing Item.

> **COMPANION CONTENT** To help you create your own management pack, an example Visual Studio project is provided as a companion file with this book (contained in the Contoso.Example.Report.Library.zip file). It will fully compile, although it does not pull any data; it is provided for example purposes to show the parts of the management pack and how they combine. Figure 7-8 shows ReportFragment.mpx open in Visual Studio. There are multiple example elements provided in the contents with descriptions and helpful hints to help you modify the fragment for your management pack.



```
<ManagementPackFragment SchemaVersion="2.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!-- This is an example Management Pack Fragment that has all of the elements you will need to
  create a custom report in Operations Manager  Multiple Examples may be provided below if you do
  not need the section you can delete it. -->
  <Reporting>

    <!-- Data Warehouse Scripts section -->
    <DataWarehouseScripts>

      <!-- Below is an example of a DataWarehouseScript section which calls a Uninstall, Install a
      <DataWarehouseScript ID="Contoso_Example_Report_Library_ExampleReport.Script" Accessibility=
        <InstallScript>Contoso_Example_Report_Library_ExampleReport_Install</InstallScript>
        <UninstallScript>Contoso_Example_Report_Library_ExampleReport_Uninstall</UninstallScript>
        <UpgradeScript>Contoso_Example_Report_Library_ExampleReport_Install</UpgradeScript>
      </DataWarehouseScript>

    </DataWarehouseScripts>
    <Reports>

      <!-- Report Section -->
```

FIGURE 7-8 ReportFragment.mpx open in Visual Studio

You'll need to modify the fragment to make it unique to your report. First configure the string resources within the Presentation element. An example string resource is provided in the management pack fragment. Each StringResource element ID must match the contents of the Prompt element within each respective report parameter. Additionally, a display string needs to be created for the string resource to ensure the control gets a friendly display name, as described earlier.

In the fragment, under the DataWarehouseScripts element, one example script is provided. Change the ID of this script to match your naming convention. Next, update the contents within the install, uninstall, and upgrade script nodes to follow your naming convention. If more than one data warehouse script is needed, for example if you have multiple reports in the management pack, you can copy the DataWarehouseScript section and update the elements as described. The contents in the InstallScript, UnInstallScript, and UpgradeScript elements need to match the ID for a corresponding Resource element found in the Resources section.

Next, configure the report parameters. The fragment includes an example of the common controls used when creating a report. Add or remove controls for your report as you see fit, making sure that you reference a string resource as described previously to ensure proper localization of the control.

Next, you need to reference the .rdl file for the actual report. Insert an ID in the ReportDefintion tag, which should correspond to a ReportReference tag in the Resources element. The ID needs to match the ID entered in the ReportDefinition tag. The file name attribute of the element needs to match the actual file name of the .rdl file created. When you build the management pack in Visual Studio and define references, Visual Studio will find and bundle them with your management pack.

Finally, define the report strings. These are the elements that the code within the .rdl file will reference to find the localized friendly display strings. Create one ReportString for each textbox or label in the .rdl file that you used an expression as defined in Listing 7-11.

Bundling the management pack

After you create the management pack, you must bundle it to include the resources. As described earlier, this is a big change between the v1.1 schema for Operations Manager 2007 R2 and the v2.0 schema for Operations Manager 2012 R2. Images, scripts, and other resources are now referenced instead of being added to the management pack as strings. Management pack bundles can contain either sealed or unsealed management packs.

MORE INFO **The scripts to create a management pack bundle can be found on TechNet at** ***http://social.technet.microsoft.com/wiki/contents/articles/15693.operations-manager-management-pack-authoring-management-pack-formats.aspx#Viewing_the_contents_of_a_management_pack_bundle*****.**

To bundle the management pack, you first need to gather all the resources that are referenced. For the example solution, you need:

- Management pack
- Report RDL
- SQL install script
- SQL uninstall script

Save the script from the site mentioned in the preceding More Info readeraid into a Windows PowerShell file called New-MPB.ps1. Place all of the parts in the same directory. Run the code in Listing 7-16.

LISTING 7-16 New-MPB.ps1 code example

```
.\New-MPB.ps1 Contoso.Example.Report.Library Contoso.Example.Report.Library
```

This will bundle all the files into a new .mpb file. At this point, your management pack is ready to be imported and all the expected files have been bundled into a single file. The expected output should look like Listing 7-17.

LISTING 7-17 New-MPB.ps1 output

```
VERBOSE: Adding MP: Contoso.Example.Reporting.Library
VERBOSE: Adding stream: Contoso.Example.Report.InstallScript
VERBOSE: Adding stream: Contoso.Example.Report.UninstallScript
VERBOSE: Adding stream: Contoso.Example.Report1.ReportRdl
VERBOSE: wrote mpb: E:\MPs\Contoso.Example.Reporting.Library.mpb
```

If you are leveraging Visual Studio Authoring Extensions to build your management pack, it is sealed and bundled automatically when you build your project; there is no need to run Windows PowerShell to bundle your management pack or FastSeal.exe tool to seal the management pack.

This page intentionally left blank

CHAPTER 8

Authoring dashboards in Power View

System Center Operations Manager has come out of the box with standard reports built on SQL Server Reporting Services for reporting since it was Microsoft Operations Manager 2005. These reports run against the data warehouse and pull historical data in the form of performance or state reports. Dashboards were introduced originally in Microsoft System Center Operations Manager 2007, but they were just views that could contain multiple other views previously configured within the main dashboard. System Center 2012 R2 Operations Manager provides more dashboard configuration options to end users. Operations Manager 2012 also includes dashboard widgets, allowing you to configure the different sections of the dashboard on demand. With subsequent updates, these customizations have expanded, allowing for even more options. Dashboard widgets certainly enhance the capability of Operations Manager, but to expand your reporting even further, you need to consider Business Intelligence (or, BI), specifically Power View.

NOTE **The examples in this chapter use PowerPivot to provide the data you need for your Power View sheets. To use Power View, whether in Microsoft SharePoint or Microsoft Excel, you must meet some specific requirements, which are described in the Power View documentation at *http://office.microsoft.com/en-us/excel-help/power-view-explore-visualize-and-present-your-data-HA102835634.aspx*.**

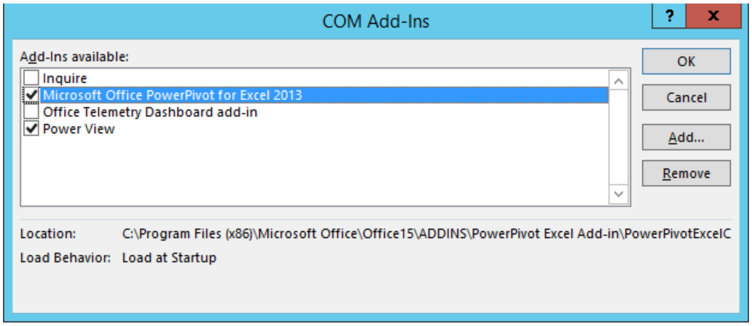
See also *For a list of all the current dashboard widgets, see http://social.technet.microsoft.com/wiki/contents/articles/24133.operations-manager-dashboard-widgets.aspx.*

Enabling Power View in Excel

By default, Power View is not enabled in Microsoft Excel 2013. To enable Power View, do the following:

1. Click File, click Options, and then click Add-Ins.

2. On the Manage drop-down list, select COM Add-Ins, and then click Go to open the COM Add-Ins dialog box:



Add-Ins available:

☐ Inquire
☑ Microsoft Office PowerPivot for Excel 2013
☐ Office Telemetry Dashboard add-in
☑ Power View

Location: C:\Program Files (x86)\Microsoft Office\Office15\ADDINS\PowerPivot Excel Add-in\PowerPivotExcelC

Load Behavior: Load at Startup

The COM Add-Ins displayed in the dialog box are extensions for Excel 2013 that allow you to do different tasks that you normally would not be able to do out of the box.

3. Select the checkboxes labeled Microsoft Office PowerPivot for Excel 2013 and Power View from the list of available add-ins, and then click OK. These two add-ins enable you to take advantage of the capabilities that PowerPivot and Power View provide.

COMPANION CONTENT Included in the companion files for this book is an Excel 2013 workbook (PowerViewDashboardExample.xlsx) that contains several example Power View dashboards for Operations Manager. This workbook contains the data model explained in the chapter. The data source needs to be updated in the workbook for it to work in your environment. To do this, open the file, click PowerPivot on the ribbon, click Manage, click Existing Connections and choose the PowerPivot connection, and then click Edit. Type your server name, choose your database, and click Save.

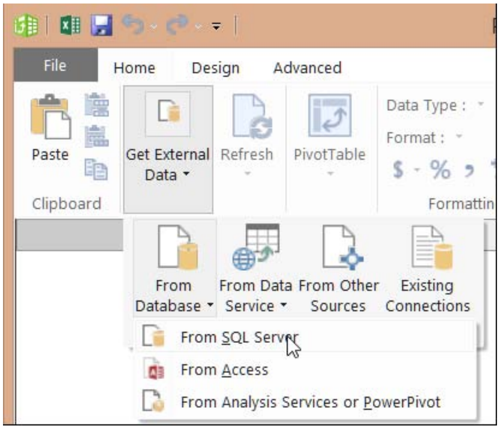
Configuring PowerPivot for your data model

The procedures in this section use PowerPivot as the data model for pulling data from Operations Manager. PowerPivot is a better choice than some of the other data source options because it is the most versatile. PowerPivot can leverage normal Excel cells, which in the case of querying the large Operations Manager data warehouse would not be plausible. Power View does provide some great functionality for leveraging tabular and multidimensional models from SQL Server Analytic Services through PowerPivot, but since the Operations Manager data warehouse does not meet either requirement, the examples in this chapter use the Operations Manager data warehouse as a standard SQL Server database.

After you launch Excel 2013 and enable the COM Add-Ins as described in the previous section, you can begin configuring PowerPivot as follows:

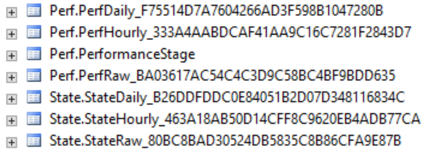
1. Click PowerPivot on the ribbon and click Manage in the Data Model section to open the PowerPivot window.

2. Click Get External Data on the ribbon, click From Database, and then click From SQL Server as shown in the following image:



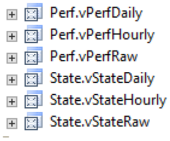
3. On the first page of the Table Import Wizard, type a friendly connection name like **Operations Manager Management Group And DW** or just **SCOM_MGMT DW**. Specifying the management group name in the connection ensures you know where your data is coming from when you have multiple management groups, such as a development, QA, and production instances of Operations Manager.
4. In the Server Name text box, type the name of your SQL database server. If you are using a named instance, you must also specify this in the Server Name text box.
5. After the server name is defined, click the Database Name combo box, select your Operations Manager data warehouse database, which is named OperationsManagerDW by default, and then click Next.
6. The next page of the Table Import Wizard presents two options: you can select the tables or views to use or you can write a query to specify the data to import. Each approach has its own benefits. For example, if you select the views you want, you can specify the relationship and have a well-structured data source, but each table or view will mirror what is in the database. If you write your own query, your data will be presented in a single table, which can help simplify your data source but will also be difficult to add to later. For the purpose of this walkthrough, you should click the Select From A List Of Tables And Views option to specify the data to import.

7. The Select Tables And Views page of the Table Import Wizard presents a grid of tables and views. It is strongly recommended that you use views instead of tables because the views abstract the tables in the database. For example, the Perf tables include one for daily, hourly, staging, and raw. These table names include GUIDs at the end as shown in the following image:



```
Perf.PerfDaily_F75514D7A7604266AD3F598B1047280B
Perf.PerfHourly_333A4AABDCAF41AA9C16C7281F2843D7
Perf.PerformanceStage
Perf.PerfRaw_BA03617AC54C4C3D9C58BC4BF9BDD635
State.StateDaily_B26DDFDDC0E84051B2D07D348116834C
State.StateHourly_463A18AB50D14CFF8C9620EB4ADB77CA
State.StateRaw_80BC8BAD30524DB5835C8B86CFA9E87B
```

However, this might change in future releases, and with other System Center databases one view could encompass multiple tables, in which case selecting a table could add unneeded overhead. The following image shows views that target the Perf tables shown above.



```
Perf.vPerfDaily
Perf.vPerfHourly
Perf.vPerfRaw
State.vStateDaily
State.vStateHourly
State.vStateRaw
```

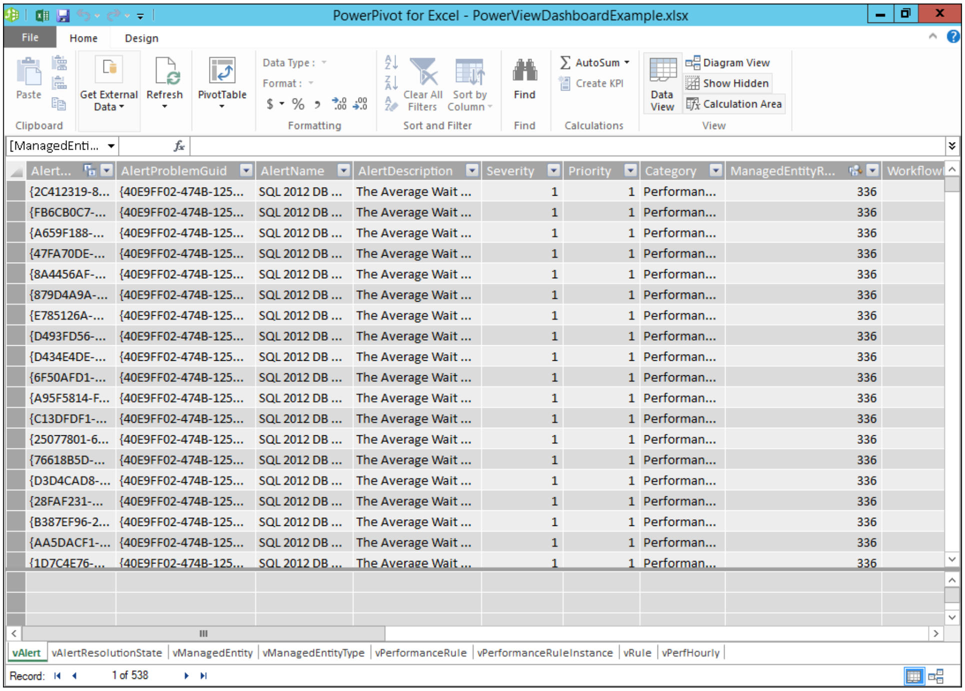
The names of these views are much cleaner and do not include GUIDs. The columns exactly match the table that the view represents. In this case the views do not do a complex join.

8. For this walkthrough, use the views shown in the following table:

VIEW	REASON
vManagedEntity	Contains the monitored objects
vManagedEntityType	Contains the types of monitored objects
vRule	Contains all rules created
vPerformanceRule	Contains performance-specific rules
vPerfHourly	Contains performance data aggregated hourly
vPerformanceRuleInstance	Contains the actual instances of the performance rules
vAlert	Contains alerts
vAlertResolutionState	Contains the resolution state of all alerts

9. In the grid on the Select Tables And Views page, select vManagedEntity, vManagedEntityType, vRule, vPerformanceRule, vPerfHourly, vPerformanceRuleInstance, vAlert, and vAlertResoutionState, and then click Finish. A message appears stating that the views were imported correctly and showing how many rows were transferred per view. Click Close to return to the PowerPivot

configuration screen. By default, the view is set to Data View, and a row of tabs along the bottom indicates all of the imported views, as shown in the following image:



Alert...	AlertProblemGuid	AlertName	AlertDescription	Severity	Priority	Category	ManagedEntityR...	Workflow
{2C412319-8...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{FB6CB0C7-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{A659F188-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{47FA70DE-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{8A4456AF-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{879D4A9A-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{E785126A-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{D493FD56-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{D434E4DE-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{6F50AFD1-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{A95F5814-F...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{C13DFDF1-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{25077801-6...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{76618B5D-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{D3D4CAD8-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{28FAF231-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{B387EF96-2...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{AA5DACF1-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	
{1D7C4E76-...	{40E9FF02-474B-125...	SQL 2012 DB ...	The Average Wait ...	1	1	Performan...	336	

Remember that when you leverage these columns in a dashboard, the title of the item will match the data source item title exactly. For example, the AlertName column will display as AlertName without a space between the two words. If you want the column title to include a space in the dashboard, you must edit it in the Data Source view. Each tab will include the data from the corresponding view that you selected. This data works just like a PivotTable: refreshing the data source updates the data.

10. At this point the data has been imported and is ready to use, but you must still configure the relationships. Defined relationships enable you to use other views to slice data on your dashboards for a much richer experience. To create the relationships, either click Create Relationship on the Design tab, or in Diagram view, drag connections between each view.

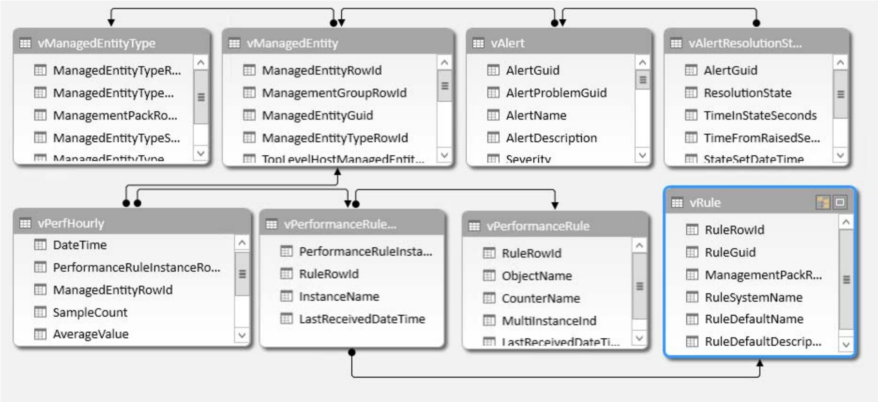
11. For this example, click Diagram View on the Design tab. Then use the information in the following table to connect the views and define their relationships.

VIEW	RELATED VIEW	COLUMN
vManagedEntity	vManagedEntityType	ManagedEntityTypeRowId
vManagedEntity	vPerfHourly	ManagedEntityRowId
vManagedEntity	vAlert	ManagedEntityRowId
vPerfHourly	vPerformanceRuleInstance	PerformanceRuleInstanceId
vPerformanceRuleInstance	vRule	RuleRowId
vPerformanceRuleInstance	vPerformanceRule	RuleRowId
vAlert	vAlertResolutionState	AlertGuid

Excel sets the type of relationship (one to many, many to many, and so on), so you don't have to set the relationship direction.

Start with the vManagedEntity view. Locate ManagedEntityTypeRowId, click and drag the column to the vManagedEntityType view and release it over the ManagedEntityTypeRowId. This will create the relationship between ManagedEntityTypeRowId in the vManagedEntity view and the ManagedEntityTypeRowId in the vManagedEntityType view. Be sure to select matching columns in each view; you may not get an error depending on the values of the columns from each view.

12. Repeat the previous step to establish all of the relationships. When you are done, your PowerPivot window should look like the following image:



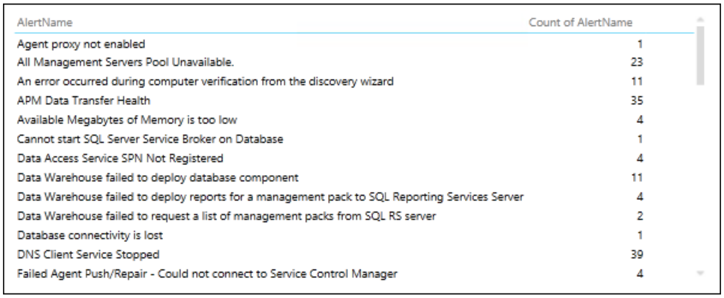
Notice that all the relationships created now have arrows pointing to the tables that they are related to. You can add additional views and create additional relationships depending on what you plan to include in your dashboard. You can also create multiple data sources, for example if you want your alert data separate from your performance or state data, you can create each as its own data source.

13. When you are done, click File, and then click Save to save your data source. Click File, and then click Close to return to the Excel workbook.

Creating a Power View dashboard

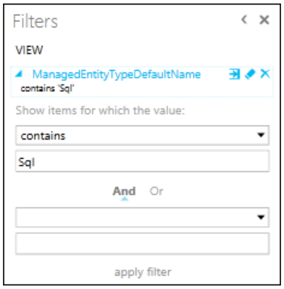
After you configure the data source you can create the Power View dashboard. You must install Silverlight or the Power View sheet will not load. To create the Power View dashboard, follow these steps:

1. In Excel, click the Insert tab, and then click Power View in the Reports section. It may take a few moments to load a blank dashboard. All of the views that you used to create your data source appear in the Power View Fields section.
2. Create an Alert Dashboard. To do this, expand vAlert in the Power View Fields section, and drag AlertName onto the Power View canvas.
3. Right-click AlertName in the Power View Fields section, and select Add To Table As Count. Your dashboard should now look similar to the following image:



AlertName	Count of AlertName
Agent proxy not enabled	1
All Management Servers Pool Unavailable.	23
An error occurred during computer verification from the discovery wizard	11
APM Data Transfer Health	35
Available Megabytes of Memory is too low	4
Cannot start SQL Server Service Broker on Database	1
Data Access Service SPN Not Registered	4
Data Warehouse failed to deploy database component	11
Data Warehouse failed to deploy reports for a management pack to SQL Reporting Services Server	4
Data Warehouse failed to request a list of management packs from SQL RS server	2
Database connectivity is lost	1
DNS Client Service Stopped	39
Failed Agent Push/Repair - Could not connect to Service Control Manager	4

4. After the core data is added, add a slicer to allow for easy manipulation of the data. The column to use for the slicer is in ManagedEntityType view, so in the Power View Fields group, expand the ManagedEntityType view and drag ManagedEntityTypeDefaultName onto the Power View canvas without releasing it on the table you defined, as shown in the previous image.
5. Drag the same ManagedEntityTypeDefaultName column to the Filters section of your canvas. To the right of ManagedEntityTypeDefaultName in the Filters section of your canvas, click the filter mode button, which looks like an arrow pointing to the right. The Filters section should look like the following image.



Filters

VIEW

ManagedEntityTypeDefaultName
contains 'Sql'

Show items for which the value:

contains

Sql

Clicking on the List filter mode button allows you to create an advanced filter. For this example, type **SQL** into the Show Items For Which The Value text box and select Contains from the drop-down list. This limits the ManagedEntityTypes to types that contain the word *SQL* in the name. Of course you could use anything else in this filter depending on what you want to include on the dashboard.

6. Click Apply Filter.

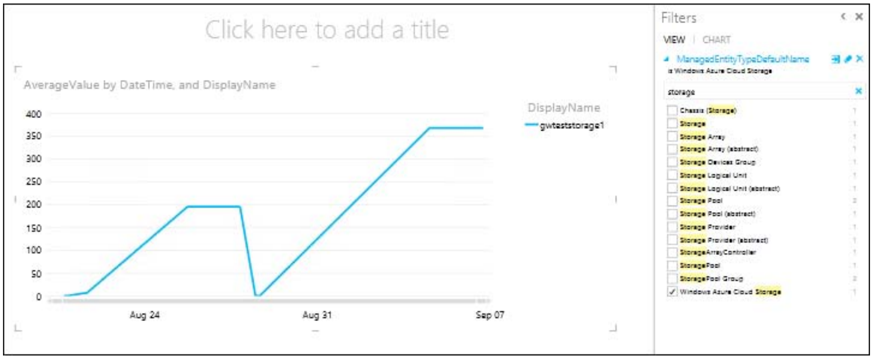
   Click the ManagedEntityTypeDefaultName section you dragged onto the canvas and, on the Design tab in Excel, click Slicer. This converts your table from a listing of objects to a selectable listing of objects, enabling you to filter your Alert table based on what you click. You can also press Shift+Click or Ctrl+Click to select multiple objects.

7. Add a title to your dashboard canvas. You have now created your first Power View dashboard for Operations Manager.

Next you will create a performance dashboard, which is very similar to the alert dashboard but slightly different. The performance dashboard will use the same data model you created, but you will be working with the performance views instead of the alert view that was used in the previous example.

1. In Excel, click the Insert tab, and then click Power View to insert a new Power View sheet.
2. Drag ManagedEntityTypeDefaultName from the vManagedEntityType view in the Power View Fields section onto the Filters section of the canvas.
3. In the ManagedEntityTypeDefaultName text box in the Filters section of the canvas, type **Windows Azure Cloud Storage,** and click the magnifying glass. You should just see one item; select the box next to it.
4. Drag AverageValue from the vPerfHourly view in the Power View Fields section to the main Power View canvas.
5. Drag DateTime from the vPerfHourly view in the Power View Fields section onto DisplayName so they merge into the same table.
6. To convert the table to a graph, with the AverageValue table selected on the Power View canvas, on the Design tab, click Other Chart, and select Line.

7. Drag DisplayName from vManagedEntity onto the newly created line graph on your Power View canvas. Your Power View canvas should now look similar to the following image:


Click here to add a title
AverageValue by DateTime, and DisplayName
400
350
300
250
200
150
100
50
0
Aug 24
Aug 31
Sep 07
DisplayName
gwteststorage1
Filters
VIEW | CHART
ManagedEntityTypeDefaultName
is Windows Azure Cloud Storage
storage
□ Chassis (Storage) 1
□ Storage 1
□ Storage Array 1
□ Storage Array (abstract) 1
□ Storage Devices Group 1
□ Storage Logical Unit 1
□ Storage Logical Unit (abstract) 1
□ Storage Pool 2
□ Storage Pool (abstract) 1
□ Storage Provider 1
□ Storage Provider (abstract) 1
□ StorageArrayController 1
□ StoragePool 1
□ StoragePool Group 2
☑ Windows Azure Cloud Storage 1

8. Give the dashboard a title and the performance dashboard for your Azure storage is complete.

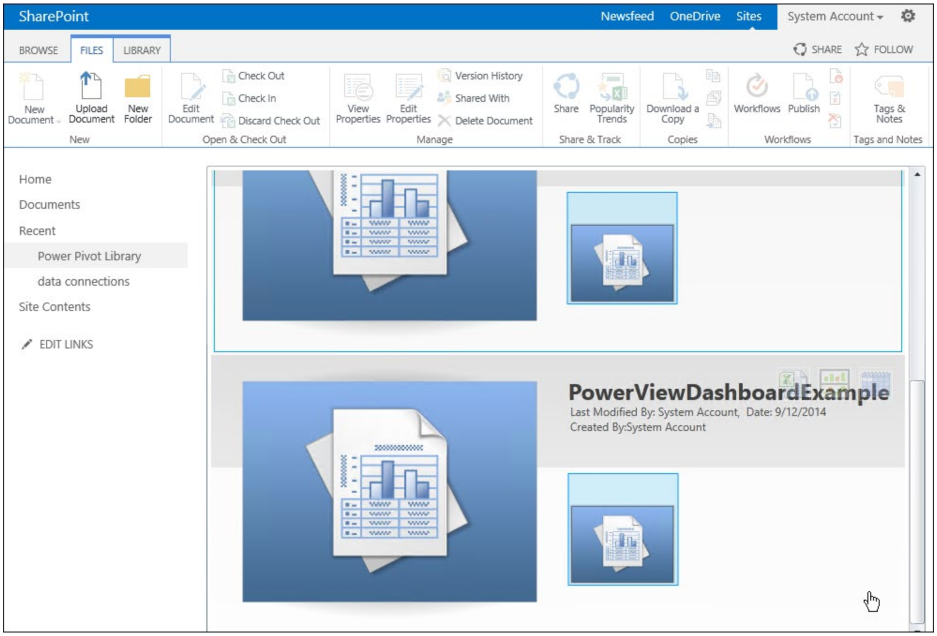
You can repeat these steps for all types of performance counters. You can also add more than only average value; for example, you can add maximum and minimum values to get a better picture of a counter that changes constantly. With Power View configurations, you can customize dashboards and display only the data you need. For example, in the performance dashboard you created above, you could also drag DisplayName onto the canvas in its own section and convert it to a slicer to show storage size for a single account.

Using Power View with SharePoint 2013

Using SharePoint 2013 to create Power View dashboards does not differ much from using Excel when your data model has already been created. But to use Power View in SharePoint 2013 you must first configure SharePoint 2013 with PowerPivot, Excel Services, Power View, and more. Instructions for configuring your SharePoint farm for this use are available at *http://technet.microsoft.com/en-us/library/jj219067.aspx*. After you configure the farm, you must create a PowerPivot library in your SharePoint site. To create this library, follow the instructions available at *http://msdn.microsoft.com/en-us/library/ee637430.aspx*.

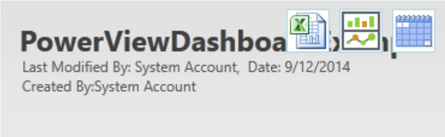
After your PowerPivot library is configured, follow these steps:

1. Open your library in your SharePoint site, click the Files tab, and then click Upload Document.
2. Browse for the Excel 2013 workbook that includes your configured PowerPivot data model, and click OK. Your site should look like the following image.



PowerViewDashboardExample
Last Modified By: System Account, Date: 9/12/2014
Created By:System Account

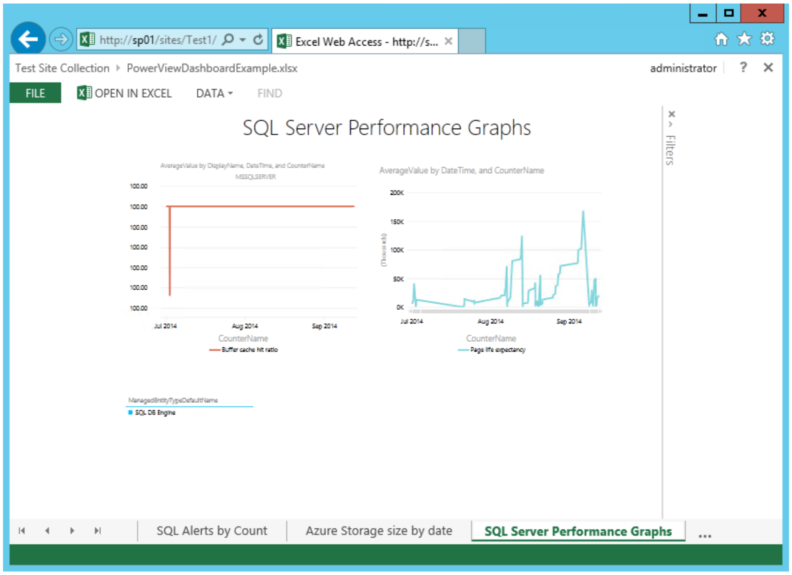
3. At this point, you can configure dashboards in one of two ways: you can open a new Excel workbook from your existing files in the PowerPivot library, or you can create a Power View Report using Reporting Services. Both methods allow you to create Power View dashboards the same way. For example, clicking the Excel icon (shown in the following image) opens a new workbook:



PowerViewDashboa
Last Modified By: System Account, Date: 9/12/2014
Created By:System Account

This workbook contains your defined data model. Clicking the Report icon next to the Excel icon launches Reporting Services, which allows you to create your dashboard in Reporting Services with the same model. Although the interfaces are slightly different, the concepts for creating dashboards are identical. Finally, clicking the calendar button allows you to schedule the data model refresh interval for the workbook.

4. As shown in the following image, dashboards created in Power View in Excel open in Excel Services, so you get all the features you expect in Excel Web Access with the ability to open it in full featured Excel.



See also *You can search many great resources online for help with configuring SharePoint, SQL Server, and Power View to visualize your data. SQL Server 2012 labs are available at http://www.microsoft.com/en-us/server-cloud/support/learning-center/virtual-labs.aspx, including one titled "Exploring Power View in SQL Server 2012 (SQL 2013)."*

This page intentionally left blank

CHAPTER 9

Troubleshooting reporting in Operations Manager

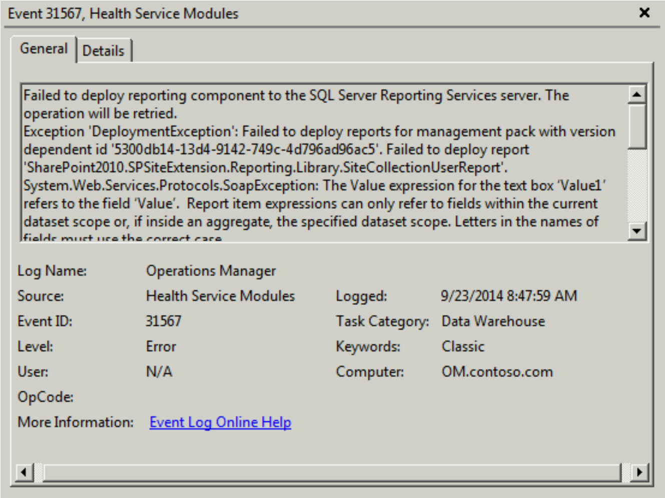
Reporting in Microsoft System Center Operations Manager is comprised of multiple parts. The data source used by the reporting feature is the Operations Manager data warehouse database. On top of the data warehouse rests SQL Server Reporting Services (SSRS), which contains all of the reports that are installed in Operations Manager that reference the Operations Manager data warehouse database. The Operations Manager SSRS instance cannot be shared with any other products, even other System Center products. Reports are deployed to SSRS through management packs, in which Operations Manager does the actual deployment to SSRS. With multiple parts, issues can arise; this chapter discusses common problems and how to resolve them.

Custom reporting issues

Custom reports built in management packs are prone to issues. During import, a management pack can pass the XML schema validation check but still have underlying issues. If you are experiencing problems with a report in a management pack, a good place to begin troubleshooting is in the OperationsManager event log of the management server. As Operations Manager attempts to deploy the report, it logs events in the OperationsManager log. These events can be used to troubleshoot many issues, not just reporting issues.

Event 31567: Failed to deploy reporting component

When you import a custom reporting management pack, you may see the event 31567 in the OperationsManager log. This event states that a reporting element failed to deploy to SSRS during report deployment. The reason for the failure will be recorded in the event details. For example, the event log in Figure 9-1 shows that the report failed to deploy because a value expression in a report text box references a field that doesn’t exist in the dataset.



```
Failed to deploy reporting component to the SQL Server Reporting Services server. The
operation will be retried.
Exception 'DeploymentException': Failed to deploy reports for management pack with version
dependent id '5300db14-13d4-9142-749c-4d796ad96ac5'. Failed to deploy report
'SharePoint2010.SPSiteExtension.Reporting.Library.SiteCollectionUserReport'.
System.Web.Services.Protocols.SoapException: The Value expression for the text box 'Value1'
refers to the field 'Value'.  Report item expressions can only refer to fields within the current
dataset scope or, if inside an aggregate, the specified dataset scope. Letters in the names of
fields must use the correct case
```

```
Log Name:          Operations Manager
Source:            Health Service Modules      Logged:          9/23/2014 8:47:59 AM
Event ID:          31567                       Task Category:   Data Warehouse
Level:             Error                       Keywords:        Classic
User:              N/A                         Computer:        OM.contoso.com
OpCode:
More Information:  Event Log Online Help
```

FIGURE 9-1 Event 31567 in the OperationsManager log

To further troubleshoot this issue, you can open the .rdl file in Business Intelligence Development Studio if you are using SQL Server 2008 R2 and below or SQL Server Data Tools if using SQL Server 2012 and above, which is installed from the SQL Server installation media. Figure 9-2 shows a report that won't import. The report table references a field called Value, but there is no Value field under the Datasets in the Report Data section. This is why the report is failing to import. The cell below the faulty Value field shows SUM(Value). This needs to be updated as well since it sums on the wrong value. If this is not corrected, the report will continue to fail to deploy. Although there could be multiple issues in your report, only the first error will be found during the deployment since the deployment fails immediately upon encountering a problem. This process may need to be repeated multiple times to successfully deploy the report.

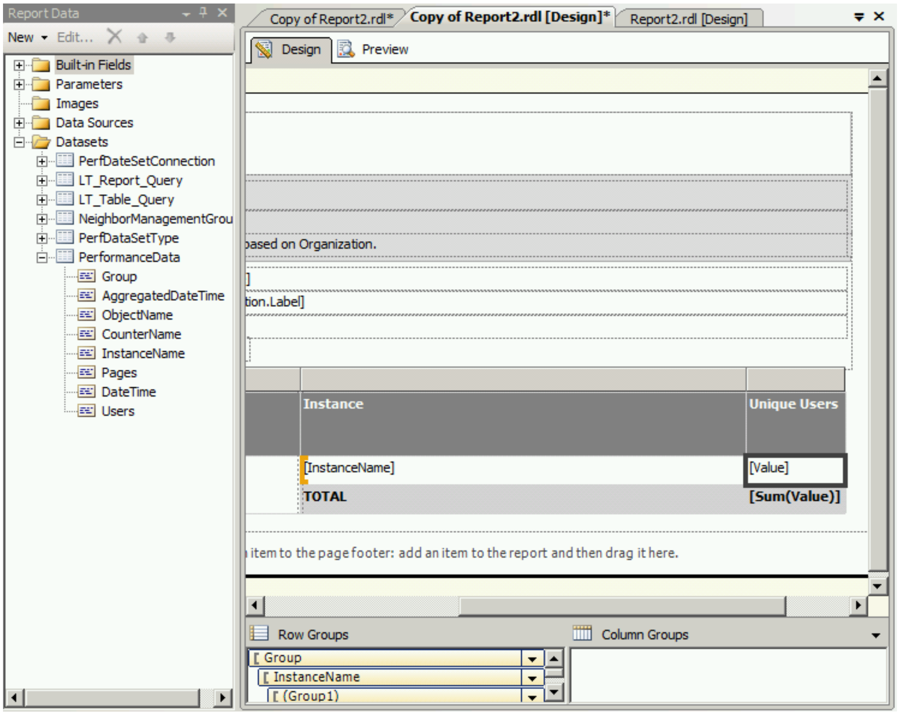
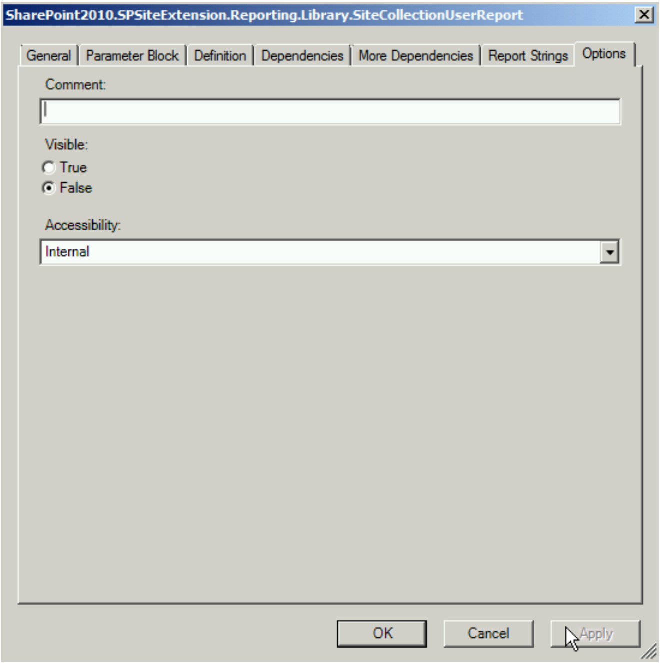


FIGURE 9-2 A report opened in Business Intelligence Development Studio

After you correct the issues with your report, you can check the OperationsManager log again to find an event 31568, which states that your report successfully deployed. This lets you know that your management pack imported correctly and the report deployed successfully. After a few minutes, the Reporting workspace in the Operations Manager console will show that the report has successfully deployed.

Report not showing in Reporting workspace

Sometimes, even though a report is successfully deployed and verified with an event 31568 in the OperationsManager log, the report is not visible in the Reporting workspace. This is another common issue with custom management packs for reports. This issue is not necessarily an error, but often happens when the Operations Manager 2007 R2 Authoring Console is used to build a management pack for reporting. In the Report Properties window shown in Figure 9-3, the Visible option is set to False. This is the default value for new reports created in the System Center Operations Manager 2007 R2 Authoring Console. Change this setting to True, reimport the management pack, and the report will be visible.



SharePoint2010.SPSiteExtension.Reporting.Library.SiteCollectionUserReport

Comment:

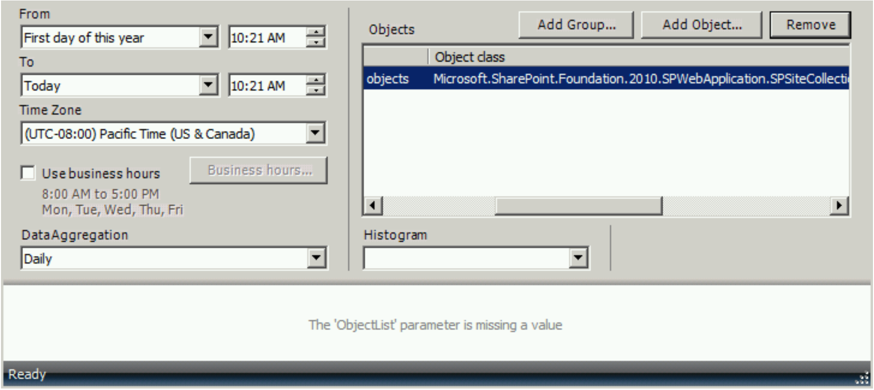
Visible:
○ True
● False

Accessibility: Internal

FIGURE 9-3 The Options tab on a report in the Operations Manager 2007 R2 Authoring Console

The ‘xxxxx’ parameter is missing a value

Because there are many parts to custom reporting, a wide array of issues can arise, including reports that import and deploy successfully but then fail during runtime. For example, you may encounter an error like “The ‘xxxxx’ parameter is missing a value,” where ‘xxxxx’ is a parameter for your report, as shown in Figure 9-4, which indicates the ‘ObjectList’ parameter is missing a value.



From: First day of this year 10:21 AM
To: Today 10:21 AM
Time Zone: (UTC-08:00) Pacific Time (US & Canada)
□ Use business hours
8:00 AM to 5:00 PM
Mon, Tue, Wed, Thu, Fri
DataAggregation: Daily

Objects

	Object class
objects	Microsoft.SharePoint.Foundation.2010.SPWebApplication.SPSiteCollecti

Histogram:

The 'ObjectList' parameter is missing a value

Ready

FIGURE 9-4 A report running from the Operations Manager console

The parameter is missing a value error can occur if the parameters defined in the parameter block section of a custom reporting management pack do not match the parameters defined in the report. You can find more information about this in Chapter 7, "Bringing it all together into a management pack," which describes the parts of the management pack that are required for reporting. In the previous example, the error occurred because the ObjectList parameter is missing a value, which means that a parameter named ObjectList in the SSRS report is not being passed a value. To troubleshoot this error, first review the ParameterBlock section of the management pack. Listing 9-1 shows the offending control in the ParameterBlock section. It reveals that this control is bound to a ReportParameter called ObjectList2, which shows that even a minor spelling mistake can cause major issues for custom reports you plan to deploy.

LISTING 9-1 Control tag with an incorrectly spelled report parameter

```
<Control
type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.MonitoringObjectXmlPi
cker" columnSpan="2">
<ReportParameters>
     <ReportParameter name="ObjectList2">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.
               DataWarehouse.Report.ParameterPrompt.ObjectList</Prompt>
     </ReportParameter>
     <ReportParameter name="ManagementGroupId" binding="GroupList" />
</ReportParameters>
     <Properties>
          <Property name="TypeFilter">

<Value>Microsoft.SharePoint.Foundation.2010.SPWebApplication.SPSite</Value>
          </Property>
     </Properties>
</Control>
```

Installation issues

In previous versions of Operations Manager, many reporting installations were corrupted due to issues encountered during the install process. In System Center 2012 R2 Operations Manager, installation of the reporting feature has been greatly improved. However, installation issues can still happen, and if an error does occur during installation, SSRS can become corrupt and unusable since Operations Manager changes the permissions to SSRS. This is why Operations Manager reporting cannot share the same SSRS instance as other products, including other System Center products. To minimize issues with SSRS, follow the installation steps on TechNet at *http://technet.microsoft.com/en-us/library/hh298611.aspx*.

An attempt to connect to the report server failed

If installing the reporting feature for Operations Manager fails and you encounter the error message "An attempt to connect to the report server failed" when you try to connect to the SSRS instance (i.e., http://<servername>/ReportServer), most likely the SSRS report instance is corrupt. To correct this corrupt installation you can use ResetSRS.exe, a tool provided with the Operations Manager installation media. Run ResetSRS.exe from the \SupportTools\<architecture>\ folder on the installation media. This will reset the permissions on SSRS and you can attempt the installation again. If resetting the permissions on SSRS does not work, you may have to uninstall and reinstall SSRS.

The specified SQL Server instance is not valid

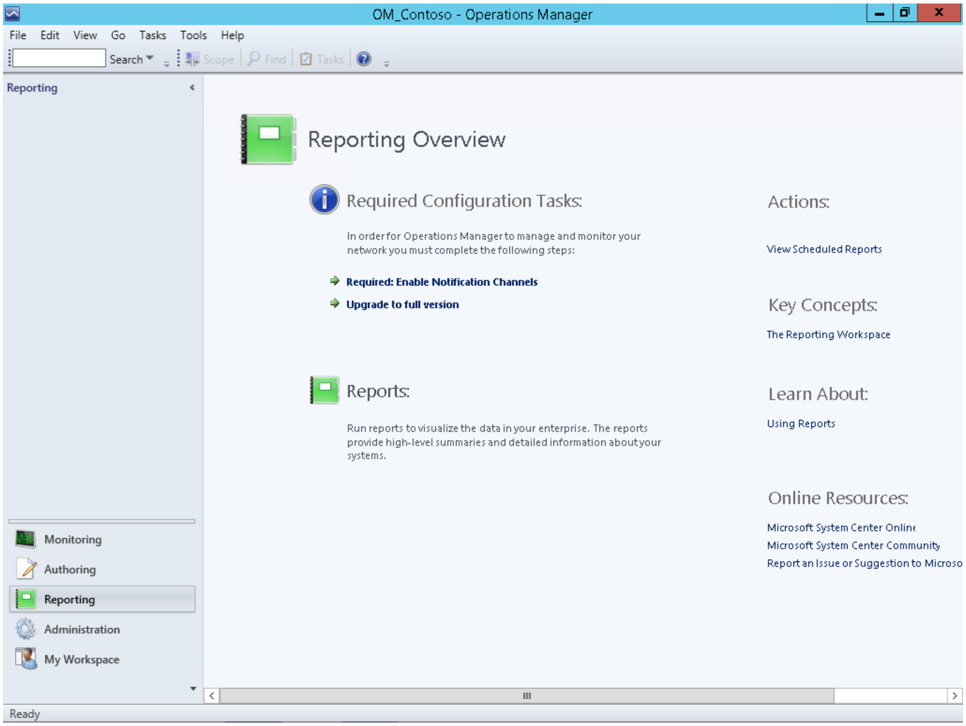
When installing reporting for Operations Manager, you may get an error indicating "The specified SQL Server instance is not valid." This most commonly happens when you attempt to install reporting for Operations Manager on an SSRS instance that is already hosting reports for another product, for example, a reporting instance that is being leveraged for System Center Configuration Manager. If the instance is already in use, you will need to stand up a new reporting instance specifically for Operations Manager. If the instance is not already in use and you encounter this error, than most likely you will need to run ResetSRS.exe as described previously to reset SSRS and re-attempt the installation.

General reporting issues

As mentioned previously, issues can arise with custom reporting creation and reporting installation. But problems can also occur during the normal use of reporting in Operations Manager. Reporting issues encountered during the use of the console can be difficult to pinpoint unless you have a good idea what you are looking for. As always, the OperationsManager log is a great place to start, although when you have a working Operations Manager management group deployed, you can also begin troubleshooting reporting issues just by looking at the console.

Reports missing from the Operations Manager console

When you launch the Operations Manager console and click on the Reporting workspace, you expect to see the console populated with all of your reports. However, sometimes when you launch the Operations Manager Operations console or the Operations Manager Web Console and navigate to the Reporting workspace, no reports are displayed, much like in Figure 9-5. The reports may be missing from the console for several reasons, and the following sections discuss some of the more common occurrences.



Reporting Overview

Required Configuration Tasks:

In order for Operations Manager to manage and monitor your network you must complete the following steps:

- Required: Enable Notification Channels
- Upgrade to full version

Reports:

Run reports to visualize the data in your enterprise. The reports provide high-level summaries and detailed information about your systems.

Actions:

View Scheduled Reports

Key Concepts:

The Reporting Workspace

Learn About:

Using Reports

Online Resources:

Microsoft System Center Online
Microsoft System Center Community
Report an Issue or Suggestion to Microso

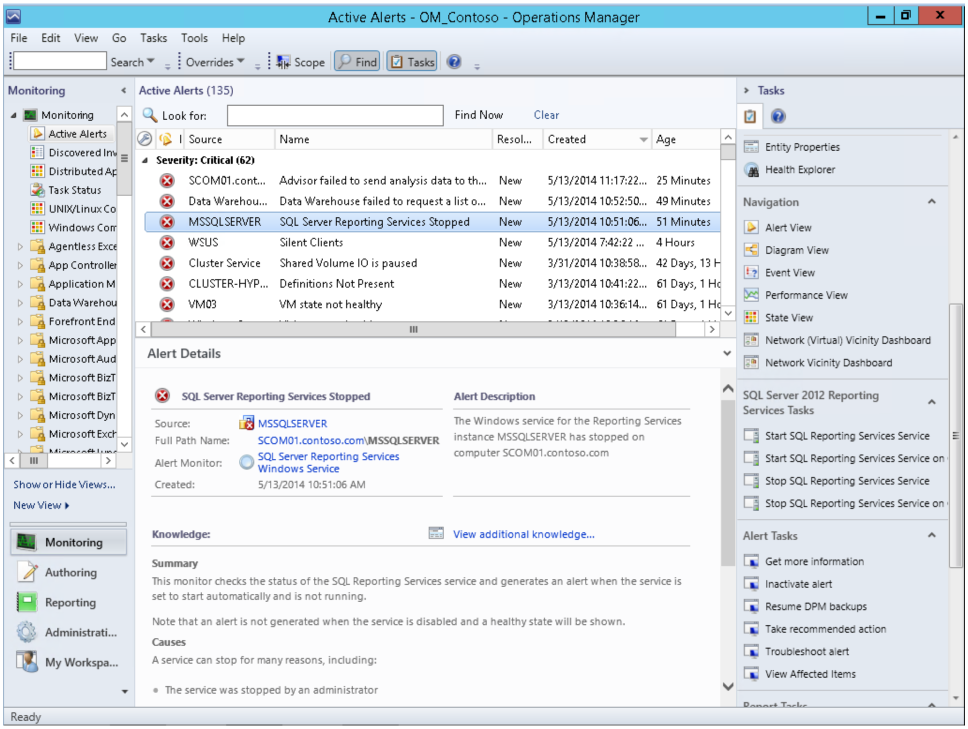
FIGURE 9-5 The Operations Manager console Reporting workspace with reports missing

SSRS Report Service is stopped

Frequently, reports are missing because the SSRS service is stopped. You can check the status of the service as you would any other Windows service, for example by using the Services MMC Snap-in, locating the SQL Server Reporting Services (Instance Name), and then viewing the status; or by using a simple PowerShell command such as Get-Service -Name ReportServer. This command will output the status of the SQL Server Reporting Services service to your command window.

You can also use SQL Server Reporting Services Configuration Manager to view the SSRS service status and even to start it or make changes to the account. However, before you make any changes to the SSRS account, you should follow the guidance found at *http://msdn.microsoft.com/en-us/library/ms160340(v=sql.110).aspx*.

If your SSRS service has been stopped for any period of time, there's a good chance you or an administrator have already been notified by Operations Manager that the service is stopped. Assuming no overrides are applied to the SQL Server Reporting Services Windows Service Alert Monitor, you should receive an alert in the Active Alerts view of the Operations console Monitoring workspace, similar to the one shown in Figure 9-6. If you scroll down in the Task pane, you'll find the option to start the SSRS service under the SQL Server 2012 Reporting Services Tasks section.



Severity: Critical (62)

Source	Name	Resol...	Created	Age
SCOM01.cont...	Advisor failed to send analysis data to th...	New	5/13/2014 11:17:22...	25 Minutes
Data Warehou...	Data Warehouse failed to request a list o...	New	5/13/2014 10:52:50...	49 Minutes
MSSQLSERVER	SQL Server Reporting Services Stopped	New	5/13/2014 10:51:06...	51 Minutes
WSUS	Silent Clients	New	5/13/2014 7:42:22 ...	4 Hours
Cluster Service	Shared Volume IO is paused	New	3/31/2014 10:38:58...	42 Days, 13 H
CLUSTER-HYP...	Definitions Not Present	New	3/13/2014 10:41:22...	61 Days, 1 Hc
VM03	VM state not healthy	New	3/13/2014 10:36:14...	61 Days, 1 Hc

SQL Server Reporting Services Stopped

Source: MSSQLSERVER
Full Path Name: SCOM01.contoso.com\MSSQLSERVER
Alert Monitor: SQL Server Reporting Services Windows Service
Created: 5/13/2014 10:51:06 AM

Alert Description

The Windows service for the Reporting Services instance MSSQLSERVER has stopped on computer SCOM01.contoso.com

Knowledge: View additional knowledge...

Summary

This monitor checks the status of the SQL Reporting Services service and generates an alert when the service is set to start automatically and is not running.

Note that an alert is not generated when the service is disabled and a healthy state will be shown.

Causes

A service can stop for many reasons, including:

- The service was stopped by an administrator

FIGURE 9-6 The Operations console highlighting the SQL Server Reporting Services Stopped alert

When you initiate the start service task, wait a few minutes for the alert to clear from the Operations console. In most cases, the alert will clear and you can then navigate to the Reporting workspace to view your reports. In some cases, you may need to close the Operations console and open it again for the reports to show up.

If you run into issues that aren't resolved by starting or restarting the service, you should investigate further. Review the System and Application Windows event logs and also verify that the account used to run the SSRS service is not disabled or locked out. If you need to change the password of the SSRS account, make sure you follow the proper procedures for modifying the Data Reader account, found at *http://technet.microsoft.com/en-us/library/hh457003.aspx*.

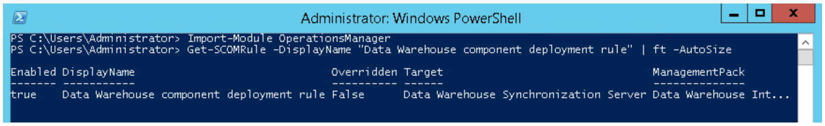
Data Warehouse Synchronization Server is missing

In some cases, simply changing the service account password or starting the SSRS service does not suffice. From time to time, you may discover that the Data Warehouse Synchronization Server entry is deleted and the reports do not get published to the report server. One way to determine if you are experiencing this is to import a new management pack that includes reports and then verify that you can see and run the existing reports in the Operations console. Depending on your infrastructure, it may take some time for the reports to show up. If the new reports do not populate the console or the report server within an acceptable amount of time, the Data Warehouse Synchronization Server entry may be deleted.

Another quick way to check for this problem is to run the following Windows PowerShell commands on one of your Operations Manager management servers.

```
Import-Module OperationsManager
Get-SCOMRule -DisplayName "Data Warehouse component deployment rule" | ft -AutoSize
```

The above commands parse the Operations Manager rules for the Data Warehouse component deployment rule. The Windows PowerShell command should return results similar to those shown in Figure 9-7. If the command does not return any data, refer to the following Microsoft Knowledge Base article for support guidance at *http://support.microsoft.com/kb/2771934*.

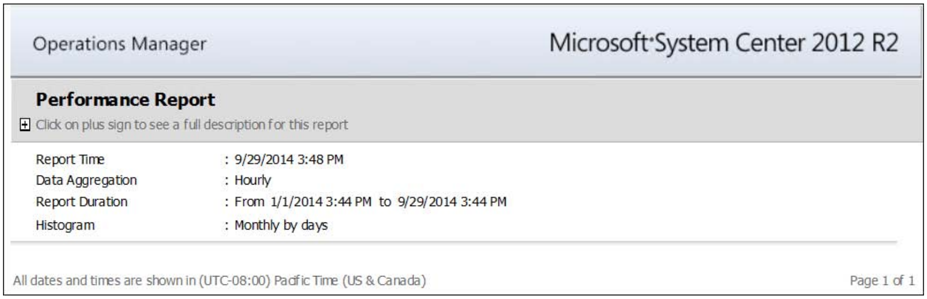


```
PS C:\Users\Administrator> Import-Module OperationsManager
PS C:\Users\Administrator> Get-SCOMRule -DisplayName "Data Warehouse component deployment rule" | ft -AutoSize

Enabled DisplayName                              Overridden Target                                ManagementPack
------- -----------                              ---------- ------                                --------------
true    Data Warehouse component deployment rule False      Data Warehouse Synchronization Server Data Warehouse Int...
```

FIGURE 9-7 The Windows PowerShell command testing for the existence of the Data warehouse component deployment rule

While not necessarily an error, a common reporting issue in Operations Manager is reports that do not contain any data. For example, Figure 9-8 shows a performance report for the current year. Data for the year does exist, but the report is empty. The most common cause of an empty report is incorrectly selected objects.



Performance Report

⊞ Click on plus sign to see a full description for this report

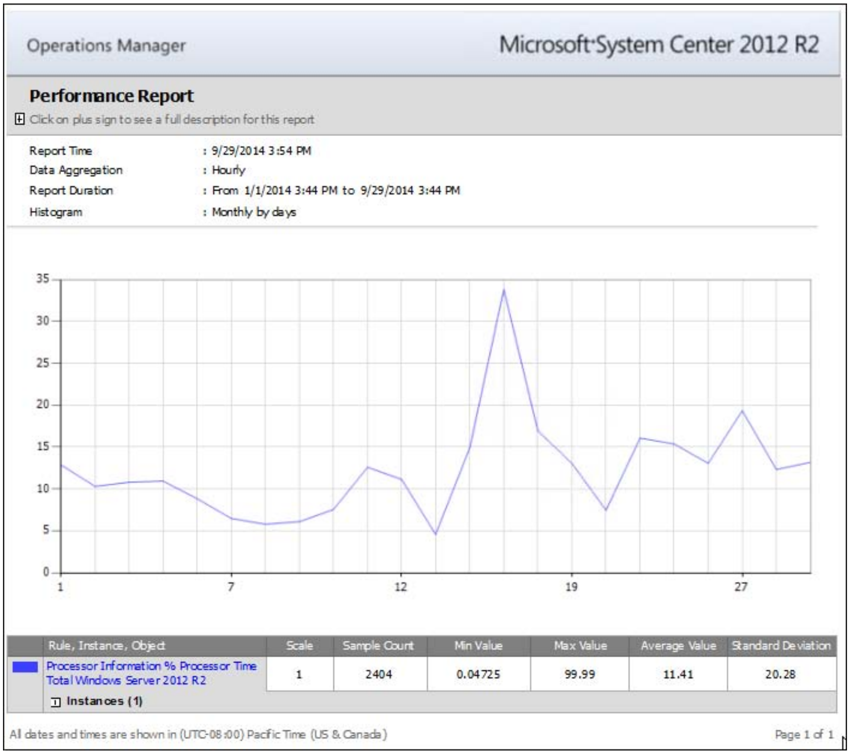
Report Time : 9/29/2014 3:48 PM
Data Aggregation : Hourly
Report Duration : From 1/1/2014 3:44 PM to 9/29/2014 3:44 PM
Histogram : Monthly by days

All dates and times are shown in (UTC-08:00) Pacific Time (US & Canada)

Page 1 of 1

FIGURE 9-8 A performance report with no data

In the example shown in Figure 9-8, the specific report chosen was the Windows Server 2012 R2 Performance History report under the Windows Server 2012 R2 Operating System (Monitoring) report folder. This report filters the objects to be displayed in the report. The description of the report indicates that it provides Total Processor % Processor Time as the performance counter. This performance counter is specific to the Windows operating system. The description of the report also specifies that the objects to be included in the report should be of the class Windows Server 2012 R2. For the example report in Figure 9-8, however, SQL Server 2012 DB Engine Group was chosen. That is the reason for the lack of data. Figure 9-9 shows the same report with Windows Server 2012 R2 Computer Group used instead; this report shows actual data.



Operations Manager

Microsoft System Center 2012 R2

Performance Report

⊞ Click on plus sign to see a full description for this report

Report Time : 9/29/2014 3:54 PM
Data Aggregation : Hourly
Report Duration : From 1/1/2014 3:44 PM to 9/29/2014 3:44 PM
Histogram : Monthly by days

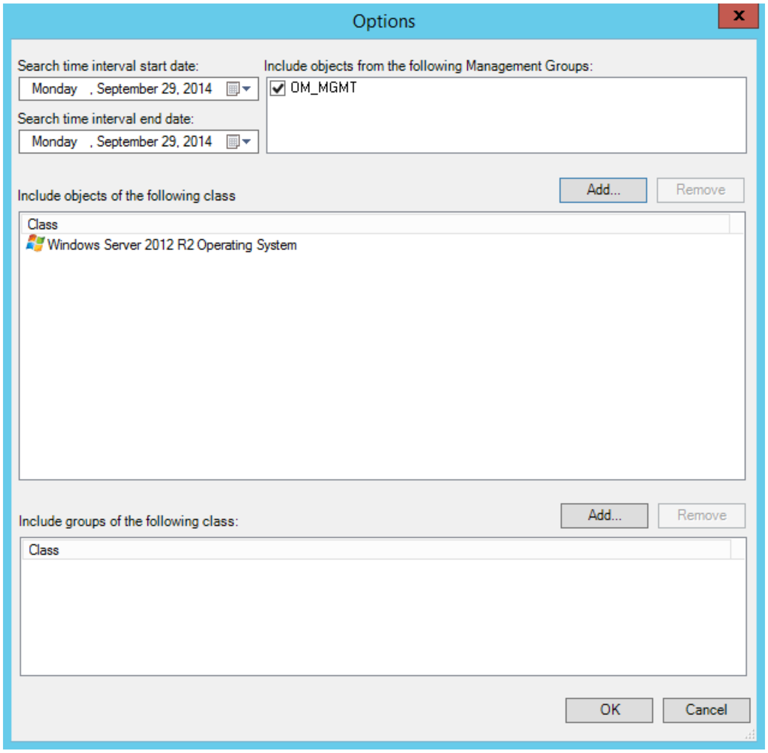
	Rule, Instance, Object	Scale	Sample Count	Min Value	Max Value	Average Value	Standard Deviation
🟦	Processor Information % Processor Time Total Windows Server 2012 R2	1	2404	0.04725	99.99	11.41	20.28
	⊞ Instances (1)						

All dates and times are shown in (UTC-08:00) Pacific Time (US & Canada)

Page 1 of 1

FIGURE 9-9 A performance report with actual data

As noted earlier, many reports filter the objects to be displayed. To determine if this is the case with a report you would like to run, open the report in the Operations console, and then choose Add Group or Add Object. You may see the notification “Filter Options have been applied” with a caution sign. Click Options to review the options that are defined for the report, as shown in Figure 9-10. In this example, Windows Server 2012 R2 Operating System is a filter applied to the objects that the report can report on. Only objects of that class will be used. If you select a different group, such as a group of SQL Server 2012 DB engines as in the earlier example, the report will be empty and not contain data.



Options

Search time interval start date: Monday , September 29, 2014

Search time interval end date: Monday , September 29, 2014

Include objects from the following Management Groups: ☑ OM_MGMT

Include objects of the following class

Class
Windows Server 2012 R2 Operating System

Include groups of the following class:

Class

FIGURE 9-10 The Options dialog box of an object selector launched from a report

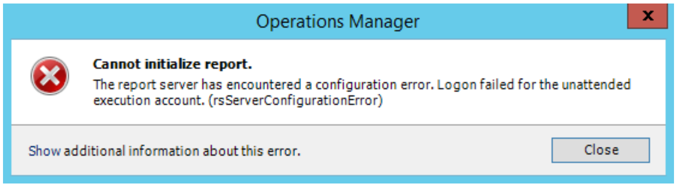
Required permissions for the Data Reader account

The Data Reader account plays a critical role in Operations Manager reporting functionalities. The Data Reader account is the account SSRS uses to run queries against the Operations Manager data warehouse database. It is a recommended practice to make this account a domain account. The Data Reader account requires logon rights to both the SQL Server nd rights to the Operations Manager management server(s).

> ***See also*** *For account-specific permissions, refer to the following TechNet blog posts: Kevin Holman's post "OpsMgr security account rights mapping - what accounts need what privileges?" at http://blogs.technet.com/b/kevinholman/archive/2008/04/15/opsmgr-security-account-rights-mapping-what-accounts-need-what-privileges.aspx and Sergio Carrilho's post "SCOM 2012 SP1 Security Accounts Matrix" at http://blogs.technet.com/b/scarrilho/archive/2013/05/02/scom-2012-sp1-security-accounts-matrix.aspx.*

Changing the password for the Data Reader account

Occasionally, you may need to change the password for one of your service accounts due to organizational policies, good practice, or an expired password. If the password has expired, you will not be able to generate reports and you will receive a logon failure message similar to the one in Figure 9-11 if you try to run a report from within the Operations console or web console.



Operations Manager

Cannot initialize report.

The report server has encountered a configuration error. Logon failed for the unattended execution account. (rsServerConfigurationError)

FIGURE 9-11 A logon failure error message from the Operations console when the password for the Data Reader Run As account is invalid

Follow these steps to change the password for the Data Reader account in Operations Manager:

1. Reset the password for the Data Reader account in Active Directory.
2. Launch the Operations Manager Operations console, navigate to the Administration workspace, then click Accounts located in the Run As Configuration section.
3. Right-click the account with the name Data Warehouse Report Deployment Account, and then click Properties.
4. When the Run As Account Properties dialog opens, click the Credentials tab, enter the new password in the Password and Confirm Password boxes, and then click OK. If your passwords do not match, you will receive a notification stating that the passwords were not confirmed. This step does not perform any account validation checks other than ensuring the passwords you entered match.

NOTE If you enter the password incorrectly or misspell the account name, you will not receive a notification saying that the account cannot be validated. This is because no account validation checks are performed when you type in here, the information is simply assumed to be correct. If the passwords do not match however, you will receive a notification here to inform you of the mismatch.

5. Launch SQL Server Reporting Services Configuration Manager. This application is typically located in the default SQL Server installation directory, Binn folder. For SQL Server 2012, the path is C:\Program Files (x86)\Microsoft SQL Server\110\Tools\Binn\RSConfig Tool.exe.
6. When SQL Server Reporting Services Configuration Manager opens, click the Execution Account tab, enter the updated password for the Data Reader account in the Password and Confirm Password boxes, and click Apply.
7. Click the Service Account tab, enter the new password, and click apply. Monitor the Results pane for the status.
8. Depending on how you configured your Report Server installation, you may have set

the Data Reader account as the Report Server Database Credential. This is the account used to connect to the report server database. If the Data Reader account was used for this, click Change Credentials, click Next, and then, on the Credentials tab, enter the Data Reader account name and the new password. Proceed through the remaining pages of the configuration wizard.

9. Open the Operations Manager Operations console and click on the Reporting workspace. You may need to refresh the workspace by pressing the F5 key. Validate that you can see and run your reports successfully. If you do not see any reports in the workspace, review any error messages and review any Event log messages for additional troubleshooting steps.

NOTE **You can perform steps 6 and 7 via the command line tool rsconfig.exe. This is located in the Binn folder provided in step 5. Guidance for using this tool can be found at** ***http://msdn.microsoft.com/en-us/library/ms162837(v=sql.110).aspx*****.**

This page intentionally left blank

About the authors

GEORGE WALLACE is a Consultant with Microsoft focusing on System Center as a whole. He has worked both on the operational side and engineering side of Operations Manager as well as the other System Center products for many years. George has a development background as well, which has led to him being involved with custom management pack development and custom reporting for Operations Manager and Service Manager as well. George regularly speaks at internal Microsoft conferences and updates his blog, which can be found on TechNet at *https://blogs.technet.com/b/georgwallace/*.

CHRIS JONES is a Senior Consultant with Microsoft. He started his IT career working as a Windows desktop engineer, followed by storage and virtualization technologies. He entered the systems management and automation arena in 2010 when he joined Microsoft and has since focused primarily on delivering System Center, App-V, and datacenter solutions to customers throughout the United States.

BILL MAY is a Senior Consultant with Microsoft Services' U.S. Public Sector organization. He has more than 16 years of IT experience, with more than 10 years at Microsoft designing and supporting System Center deployments throughout the United States region. Bill has a deep knowledge of datacenter operations and specializes in infrastructure and application monitoring solutions. He has been working with Operations Manager since MOM 2000 and loves to recall his glory days of supporting SQL Server 6.5 back when the cloud was called the "world wide web."

FRED LEE is a Premier Field Engineer with Microsoft. After supporting Operations Manager as a Support Escalation Engineer for 5 years, he moved to the Premier Field Engineer role, continuing his support in the System Center space. His primary areas of focus are Operations Manager, Orchestrator, and Service Manager.

About the series editor

MITCH TULLOCH is a well-known expert on Windows Server administration and virtualization. He has published hundreds of articles on a wide variety of technology sites and has written, contributed to, or been series editor for over 50 books. His free ebooks *Introducing Windows Server 2012* (Microsoft Press, 2012) *and Introducing Windows Server 2012 R2* (Microsoft Press, 2013) have together been downloaded over a million and a quarter times! For a complete list of free ebooks from Microsoft Press, visit the Microsoft Virtual Academy at *http://www.microsoftvirtualacademy.com/ebooks*.

Mitch has repeatedly received the Microsoft Most Valuable Professional (MVP) award for his outstanding contributions to supporting the global IT community. He is a ten-time MVP in the technology area of Windows Server Software Packaging, Deployment & Servicing. You can find his MVP Profile page at *http://mvp.microsoft.com/en-us/mvp/Mitch%20Tulloch-21182*.

Mitch is also Senior Editor of WServerNews, a weekly newsletter focused on system administration and security issues for the Windows Server platform. With almost 100,000 IT pro subscribers worldwide, WServerNews is the largest and oldest-running Windows Server–focused newsletter in the world. Visit *http://www.wservernews.com* and subscribe to WServerNews today!

Mitch runs an IT content development business based in Winnipeg, Canada, that produces white papers and other collateral for the business decision maker (BDM) and technical decision maker (TDM) audiences. His published content ranges from white papers about Microsoft cloud technologies to reviews of third-party products designed for the Windows Server platform. Before starting his own business in 1998, Mitch worked as a Microsoft Certified Trainer (MCT) for Productivity Point.

For more information about Mitch, visit his website at *http://www.mtit.com* or follow Mitch on Twitter @mitchtulloch.