



Microsoft System Center

Configuration Manager Field Experience

Rushi Faldu ▪ Manish Raval ▪ Brandon Linton ▪ Kaushal Pandey
Mitch Tulloch, Series Editor

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

Copyright © 2013 by Microsoft Corporation (All)

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Control Number: 2013950970
ISBN: 978-0-7356-8304-4

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 mspininput@microsoft.com. Please tell us what you think of this book at <http://www.microsoft.com/learning/booksurvey>.

Microsoft and the trademarks listed at <http://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Anne Hamilton
Developmental Editor: Karen Szall
Project Editor: Karen Szall
Editorial Production: Megan Smith-Creed
Copyeditor: Megan Smith-Creed
Cover Illustration: Twist Creative, Seattle
Cover Design: Microsoft Press Brand Team

Contents

<i>Foreword</i>	<i>vii</i>
<i>Introduction</i>	<i>ix</i>

PART I WMI FOR SYSTEM CENTER 2012 CONFIGURATION MANAGER ADMINISTRATORS

Chapter 1 Introduction to WMI in Configuration Manager 2012	3
Web-Based Enterprise Management (WBEM)	3
Windows Management Instrumentation (WMI)	4
Managed objects and WMI providers	4
WMI infrastructure	5
WMI consumers	6
Understanding WMI schema	6
Namespaces	7
Classes	7
Instances	7
Using WMI tools to explore WMI namespaces and classes.....	7
Configuration Manager 2012 specific WMI namespaces, classes, properties, and methods	9
Using WMI classes to create Configuration Manager queries and dynamic collections.....	11
Interaction between the site server and the Configuration Manager client	17
Reading from the SMS Provider, WMI Scripting, and the Configuration Manager SDK	19

What do you think of this book? We want to hear from you!
Microsoft is interested in hearing your feedback so we can continually improve our
books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Chapter 2 Configuration Manager custom reporting	23
Configuration Manager SQL database design.....	23
Configuration Manager database tables and views	24
The difference between SQL tables and views	26
Views	26
Tables	27
Common Configuration Manager views.....	27
Working with SQL inner and left outer joins	28
PART II SYSTEM CENTER 2012 CONFIGURATION MANAGER REPORTING	
Chapter 3 Integrating SQL Server Reporting Services with Configuration Manager 2012	33
Prerequisites for reporting services points in Configuration Manager 2012	33
Installing a Reporting services point.....	35
Chapter 4 Customizing SSRS reports for Configuration Manager 2012	39
Chapter 5 Customizing function-based built-in reports	45
PART III CONFIGURATION MANAGER DEPLOYMENT TIPS	
Chapter 6 Operating system deployment tips	57
Boot images.....	57
Enabling F8 command prompt support	57
Boot image driver management	58
Optional components	59
Adding Windows PE 3.1 to Configuration Manager 2012 SP1 CU2.....	60
Drivers.....	64
Driver signing	64
Driver maintenance	64

UEFI	65
Operating System Images and Offline Servicing.....	66
Task sequences	68
MDT integration	68
Static IP address assignment	69
OSDPreserveDriveLetter	71
SMSTSPostAction	72
Extending task sequence logging	72
Applications.....	73
Chassis type global conditions	73
Installing applications that require interaction	74
Application logging	74
Troubleshooting	74
CMTrace	75
Log locations	76
Centralized logging	77
<i>About the Authors</i>	79
<i>About the Series Editor</i>	81

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

This page left intentionally blank.

Foreword

Ever since the client-server computing architecture became mainstream, IT pros around the world have been challenged and required to manage these servers and clients. As more client computers were introduced in IT environments and started playing a critical role in performing day-to-day tasks, the need to manage them became even more urgent. More importantly, these clients became an integral part of any business's productivity and started to perform more mission-critical tasks.

Today, the clients are becoming more powerful, smarter, and increasingly mobile. They have now become assets. As these assets grow in number, become more portable, and store critical business data, the risk to organizations increases. Now, more than ever before, there is a need for IT pros to manage, monitor, and secure these assets.

Windows Active Directory and Group Policy were the starting points for IT pros to secure some aspects of these assets. However, they weren't sufficient and didn't give IT pros the ability to manage the lifecycle of these assets.

In 1994, Microsoft introduced Systems Management Server (SMS) 1.0. It was the beginning of client management solution, but more in the non-Active Directory era. SMS 2003 truly ushered in an era of advanced client management that leveraged Active Directory and all of its functionality. The adoption and popularity of SMS has continued to grow since SMS 2003, and Microsoft has pushed the limits of the solution and its ability over time.

Microsoft System Center Configuration Manager 2007 changed the game with the vision of an integrated solution along with other System Center products. Microsoft introduced many new features and firsts with Configuration Manager 2007 and took client management to a whole new level with System Center 2012 Configuration Manager. Now, Configuration Manager (both 2007 and 2012) is now an integral part of the IT infrastructure of many companies, and expertise with Configuration Manager has become one of the most sought after IT skills around the globe.

Microsoft Press and the authors of this ebook have a passion for helping IT pros working with Configuration Manager enhance their knowledge and make the most of the solution. The authors of this ebook are Microsoft Consultants from Microsoft Consulting Services (MCS) and Premier Field Engineers (PFE) from Microsoft Global Business Support (GBS) organizations with real field experience. The authors have come together to share their collective knowledge and experiences from both consulting and support in the field.

The authors have identified and chosen topics that are used on a daily basis by all Configuration Manager 2012 administrators around the world irrespective of the size and complexity of the solution or the industry it is deployed in. The authors have made an attempt to cover topics that are usually pain points for most Configuration Manager administrators. The authors have broken these into two ebooks: *System Center: Configuration Manager Field Experience* and *System Center: Troubleshooting Configuration Manager*.

We hope you enjoy this ebook and the other one as much as the authors have enjoyed writing them, and that these resources help make the most of your System Center 2012 Configuration Manager solution.

Manish Raval

Consultant, Microsoft Consulting Services (MCS)

Introduction

The purpose of this book is to provide System Center 2012 Configuration Manager administrators with helpful and tested real-world guidance from consultants and product experts at Microsoft. We want you to get the most out of using Configuration Manager in your environment regardless of whether the task at hand is querying the Configuration Manager database for system information, creating and customizing reports, or deploying operating system images to client machines.

The book is divided into the following three parts:

- *Part 1: WMI for System Center Configuration Manager 2012 administrators* is designed to familiarize you with basic WMI concepts and tools, show you how to use WMI classes to create Configuration Manager queries, and help you understand the underlying database design of Configuration Manager so you can create custom reports for querying the information you need.
- *Part 2: System Center 2012 Configuration Manager Reporting* is designed to familiarize you with SSRS and show you how to integrate SSRS with Configuration Manager 2012 and to customize and create new reports using SSRS.
- *Part 3: Configuration Manager Deployment Tips* provides tips that cover a wide variety of different scenarios to help you ensure the success of your deployments.

The target audience for this book is administrators who have at least three years experienced working with previous versions of Configuration Manager and who have begun deploying and using Configuration Manager 2012 in their environment.

About the companion content

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

<http://aka.ms/SCconfigFE/files>

The companion content includes the following:

- The sample VBScript script in Chapter 1
- The T-SQL query in Chapter 4

Errata & book support

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

<http://aka.ms/SCconfigFE/errata>

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

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

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

We want to hear from you

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

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

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

Stay in touch

Let's keep the conversation going! We're on Twitter: *<http://twitter.com/Microsoft-Press>*.

PART I

WMI for System Center 2012 Configuration Manager administrators

In Configuration Manager, Windows Management Instrumentation (WMI) plays a major role in storing site-specific and client-related management information. The two chapters in Part I of this book are designed to familiarize you with basic WMI concepts and tools, show you how to use WMI classes to create Configuration Manager queries, and help you understand the underlying database design of Configuration Manager so you can create custom reports for querying the information you need.

This page left intentionally blank.

CHAPTER 1

Introduction to WMI in Configuration Manager 2012

In Microsoft System Center 2012 Configuration Manager, Windows Management Instrumentation (WMI) plays a major role in storing site-specific and client-related management information. WMI is the Microsoft implementation of Web-Based Enterprise Management (WBEM) and Common Information Model (CIM) standards from the Distributed Management Task Force (DMTF).

The purpose of WMI is to define a proprietary set of environment-independent specifications that allow management information to be shared between management applications. WMI prescribes enterprise management standards and related technologies for Windows operating systems that work with existing management standards, such as Desktop Management Interface (DMI) and Simple Network Management Protocol (SNMP). WMI complements these other standards by providing a uniform model. This model represents the managed environment through which management data from any source can be accessed in a common way. WMI is also the key component of System Center 2012 Configuration Manager. Management information related to site server and client is stored within WMI.

This chapter describes WMI, how the WMI tools interact with WMI, and the namespaces and classes specific to Configuration Manager 2012. This chapter builds a strong foundation for understanding WMI.

Web-Based Enterprise Management (WBEM)

Web-Based Enterprise Management (WBEM) is the industry joint effort to provide a unifying mechanism for describing and sharing management information. Prior to WBEM, administrators of systems running on multiple platforms did not have any easy way to obtain management data from their different platforms. They had to use individual application programming interfaces (APIs) or a separate console for each management application. WBEM defines a common mechanism for sharing management information, and provides a consistent view of managed environments without locking you into one management framework, protocol, or platform.

WBEM was developed by the Desktop Management Task Force (DMTF). The two main goals motivating the DMTF to create WMEB are:

- To standardize the publication of management data
- To provide a standard method for accessing management information

To standardize the publication of management data, the DMTF adopted a standardized data model called the Common Information Model (CIM). CIM is an object-oriented schema for describing a system's management objects. It offers one extensible data description mechanism for all enterprise systems, network devices, and other management tools such as applications, peripherals, and databases. CIM supports data inheritance and associations and is independent of any execution environment or programming language model.

Previously, administrators had to use customized API calls and software designed specifically for each environment that they wanted to access management data from. WBEM provides one method for accessing management data that originates from disparate sources.

Windows Management Instrumentation (WMI)

WMI provides a uniform interface for any local or remote applications or scripts that obtain management data from a computer system, a network, or an enterprise. The uniform interface is designed such that WMI client applications and scripts do not have to call a wide variety of operating system APIs. Many APIs cannot be called by automation clients like scripts or Visual Basic applications. Other APIs do not make calls to remote computers. Figure 1-1 shows the WMI architecture.

Managed objects and WMI providers

A WMI provider is a COM object that monitors one or more managed objects for WMI. A managed object is a logical or physical enterprise component, such as a hard disk drive, network adapter, database system, operating system, process, or service.

Similar to a driver, a provider supplies WMI with data from a managed object and handles messages from WMI to the managed object. WMI providers consist of a DLL file and a Managed Object Format (MOF) file that defines the classes for which the provider returns data and performs operations. Providers, like WMI C++ applications, use the COM API for WMI.

For example, the preinstalled Registry provider, which accesses data in the system registry, has one WMI class, StdRegProv, with many methods but no properties. Other preinstalled providers, such as the Win32 provider, usually have classes with many properties but few methods, such as Win32_Process or Win32_LogicalDisk. The Registry provider DLL file, Stdprov.dll, contains the code that dynamically returns data when requested by client scripts or applications.

For X86 versions of Windows, WMI MOF and DLL files are located in %WINDIR%\System32\Wbem. For x64 versions of Windows, these files are located in %WINDIR%\Sys-WOW64\Wbem together with the WMI Command-Line Tools, such as Winmgmt.exe and Mofcomp.exe. Provider classes, such as Win32_LogicalDisk, are defined in MOF files, and then compiled into the WMI repository at system startup.

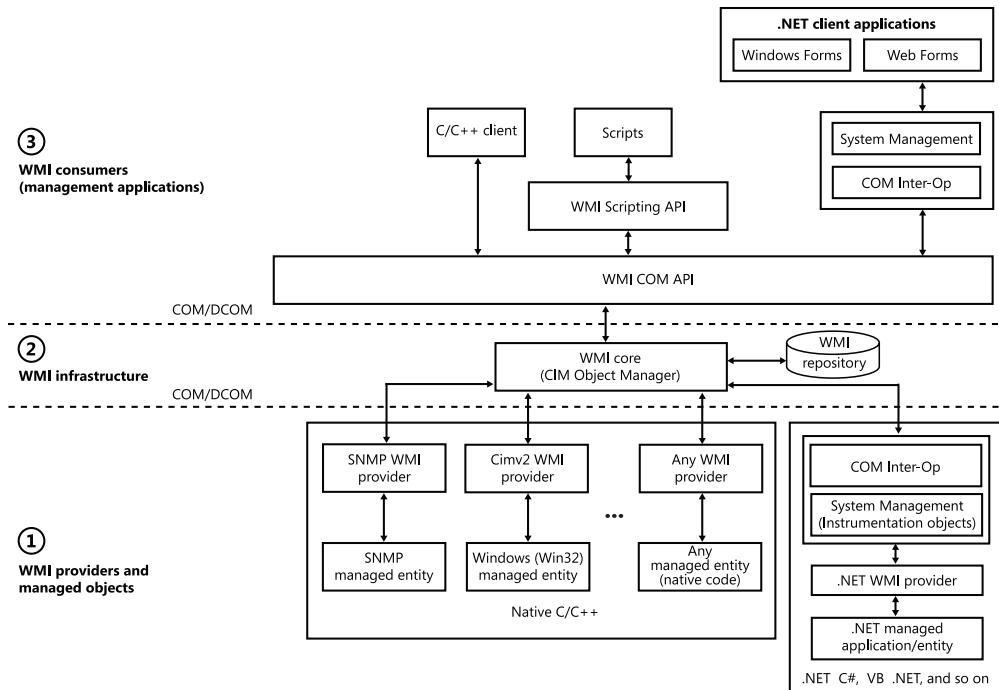


FIGURE 1-1 WMI architecture.

WMI infrastructure

The WMI infrastructure is a Microsoft Windows operating system component. The WMI infrastructure is implemented by the WMI service (`winmgmt`) and includes the WMI core and the WMI repository.

The WMI repository is organized by WMI namespaces. The WMI service creates some namespaces such as `root\default`, `root\cimv2`, and `root\subscription` at system startup and preinstalls a default set of class definitions, including the Win32 classes, the WMI system classes, and others. The remaining namespaces found on a system are created by providers for other parts of the operating system or products.

The WMI service acts as an intermediary between the providers, management applications, and the WMI repository. Only static data about objects is stored in the repository, such as the classes defined by providers. WMI obtains most data dynamically from the provider when a client requests it. You also can set up subscriptions to receive event notifications from a provider.

WMI consumers

A WMI consumer is a management application or script that interacts with the WMI infrastructure. A management application can query, enumerate data, run provider methods, or subscribe to events by calling either the COM API for WMI or the Scripting API for WMI. The only data or actions available for a managed object, such as a disk drive or a service, are those that a provider supplies.

Understanding WMI schema

Whereas the WMI object model defines how programs work with WMI, the WMI schemas define the actual implementation of WMI objects. Consider an analogy of a driving manual versus a map. A driving manual explains the techniques of driving a car; a map illustrates where the destinations are and how to get to them. The object model is analogous to a driving manual, and schemas are like maps. Understanding WMI schemas allows you to understand the relationships among the objects that WMI manages.

The DMTF defines a standard schema for WBEM called the CIM schema. This schema is implemented as the Cimv2 namespace in WMI. The CIM schema, in the form of the core and common models, provides a conceptual architecture for a managed environment. It is a framework of organizing principles that can be used by schema designers to understand and analyze the information requirements of management applications. The common model is represented by a set of abstract and concrete classes that define the basic characteristics of systems, networks, applications, and various groupings of statistical and other computer management-related data.

Figure 1-2 shows the relationship between the namespace, class, properties, method, and instance as explained above.

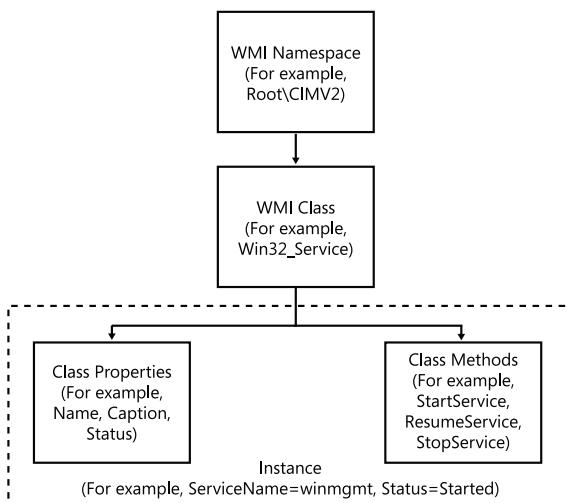


FIGURE 1-2 Namespace, classes, and instance relationship.

Namespaces

Namespaces contain classes and instances. Namespaces are not physical locations; they are more like logical databases. Namespaces can be nested. That is, within a namespace, there can be other namespaces that define subsets of objects.

CIM classes are organized into namespaces. Namespaces are the partitioning mechanism employed by the CIM to control the scope and visibility of managed resource class definitions. Each namespace in the CIM contains a logical group of related classes representing a specific technology or area of management.

Namespaces are roughly equivalent to folders on a disk drive. Like folders, namespaces provide a place to store related information; a folder named Scripts is likely to contain scripts and a namespace named MicrosoftActiveDirectory is likely to contain WMI classes used to manage Active Directory. Both folders and namespaces help you to uniquely identify an item. You can have only one file on a computer named C:\Scripts\WMI_Script.vbs; likewise, you can have only one WMI class named root\cimv2:Win32_Process.

One difference between folders and WMI namespaces is that folders are often deeply nested; for example, it is common to have folders such as C:\Program Files\Microsoft Office\Office\Office15. By contrast, namespaces rarely go more than three levels deep; the vast majority of classes useful in system administration scripts reside in the root\cimv2 namespace, a namespace nested only two levels deep.

Classes

The namespace contains the classes as explained above. All classes within a namespace must have a unique class name, and classes in one namespace cannot be derived from classes in another namespace. This is why you will find identical system, core, and common classes defined in multiple namespaces.

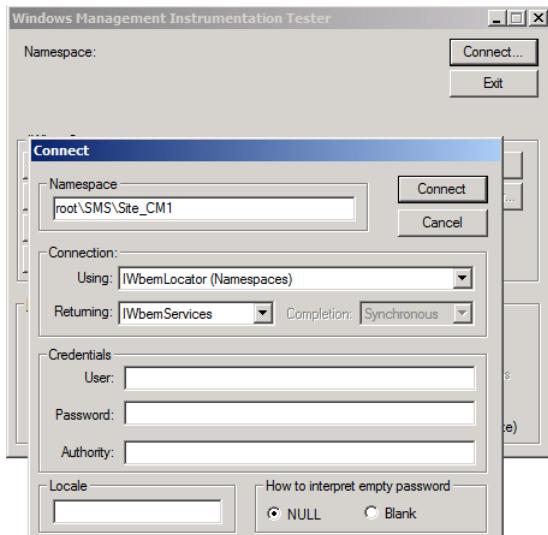
Instances

An instance is a particular manifestation of a class. Instances are more commonly thought of as data. Because instances are objects, the two terms are often used interchangeably. However, instances are usually thought of in the context of a particular class, whereas objects can be of any class.

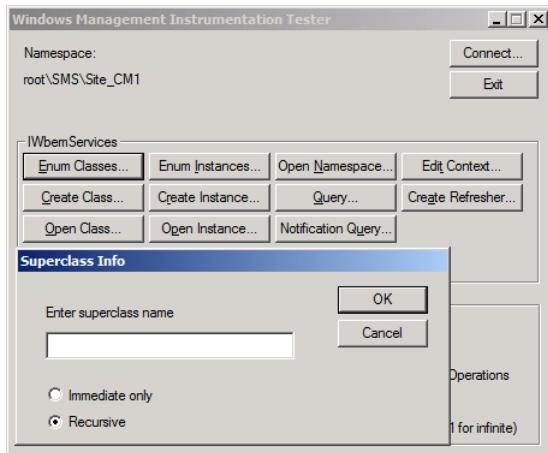
Using WMI tools to explore WMI namespaces and classes

To access the WMI, Windows operating systems include the built-in tool called the Windows Management Instrumentation Tester (Wbemtest.exe). By using the Wbemtest, you can navigate the different WMI classes and see the different properties and methods. The following procedure demonstrates the use of Wbemtest for browsing Configuration Manager classes.

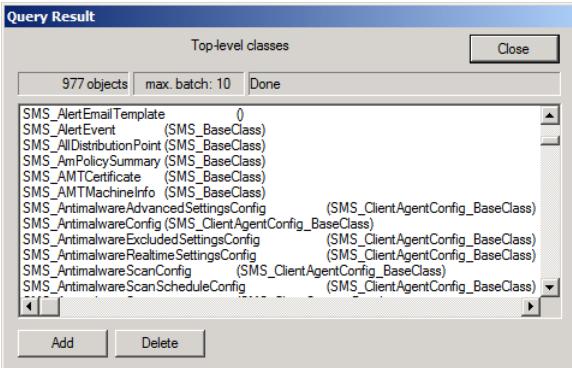
1. Press the Windows logo key + R and type **Wbemtest.exe** to launch the Windows Management Instrumentation Tester.
2. Click Connect and type **root\SMS\Site_CM1** to specify the namespace.



3. Click Connect again and, when connected, click Enum Classes to open the Superclass Info dialog box and select Recursive:



4. Click OK to return a large number of Configuration Manager classes as shown in the following screenshot:



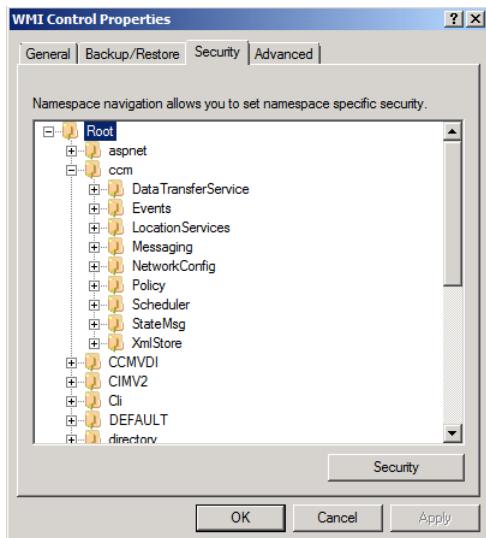
NOTE By using the Wbemtest you can navigate any WMI class and access its properties and methods.

Configuration Manager 2012 specific WMI namespaces, classes, properties, and methods

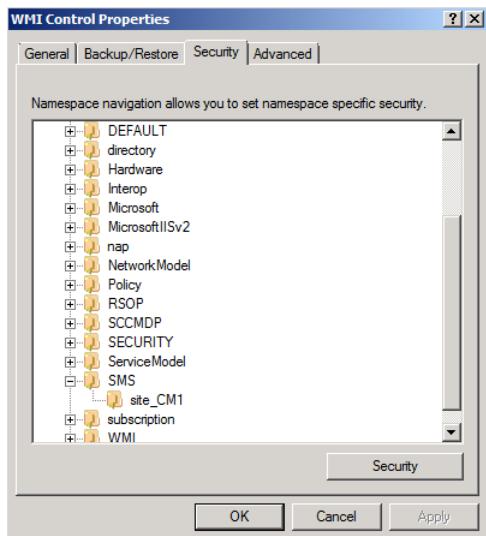
When Configuration Manager 2012 is installed on a site server, several namespaces and a number of different WMI classes are added to the WMI of the site server. This section explores some of the different WMI namespaces and WMI classes specific to Configuration Manager 2012. Note that each site might have classes that other sites might not have depending on the specific site settings, the inventory that is tracked, and so on.

To view the Configuration Manager 2012 specific namespaces, complete the following steps on the Configuration Manager 2012 site server.

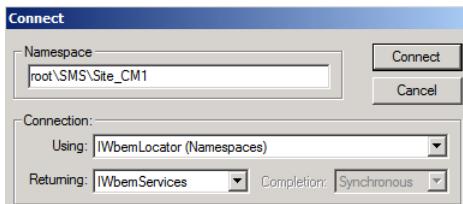
1. Open the Computer Management console and, under Services And Applications, right-click WMI Control and then select Properties.
2. Click the Security tab to view all the different namespaces. For example, the namespace root\CCM is specific to the Configuration Manager 2012 client, as shown in the following screenshot:



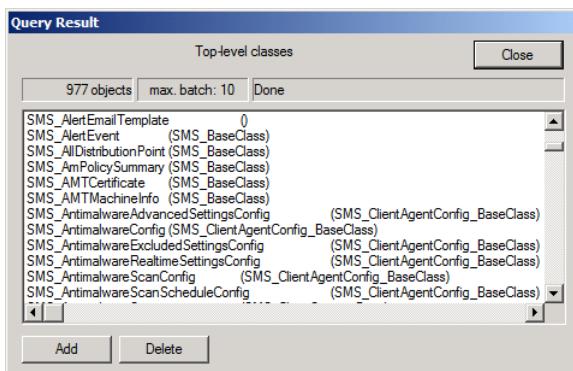
And the namespace root\SMS\site_<siteCode> is specific to the Configuration Manager 2012 site server, as shown in the following screenshot.



3. Launch Wbemtest on the site server and specify the namespace as root\SMS\site_<siteCode>. and click Connect



4. Click Enum Classes to open the Superclass Info dialog box, select Recursive, and click OK. You will see several classes whose names begin with SMS_. All of these classes are specific to the site server and are created during installation of Configuration Manager 2012.



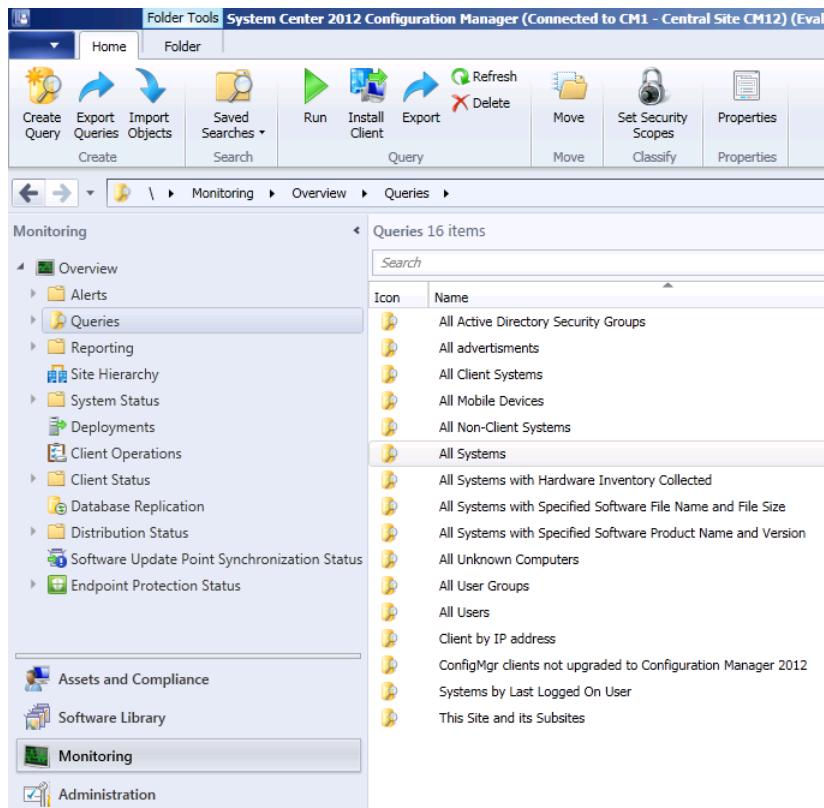
Using WMI classes to create Configuration Manager queries and dynamic collections

In Configuration Manager 2012, use queries to list the resources based on specific criteria. The Configuration Manager Queries node and collections use a WMI Query Language (WQL) query to request the Configuration Manager Object data from the SMS provider WMI schema, which in turn retrieves the data from the site database. To retrieve the data from WMI repository, you need to be able to create queries using WQL.

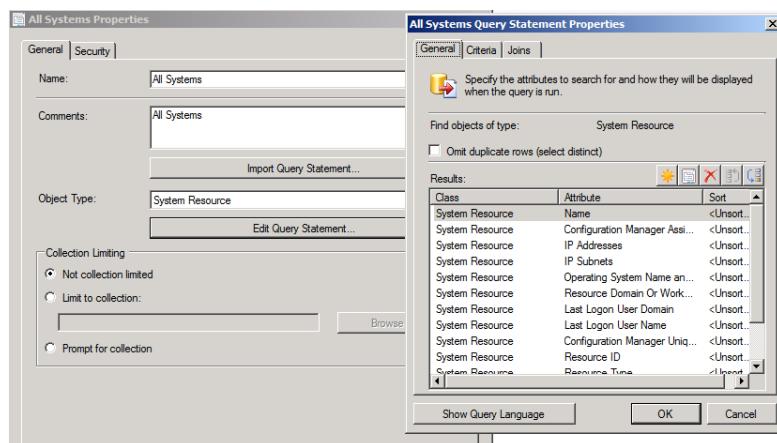
NOTE WQL is also sometimes referred to as SQL for WMI.

In the previous section you saw that several WMI classes are created during installation of the Configuration Manager 2012 site server. In this section, you start with the built-in Configuration Manager queries to view the respective WMI classes and properties used to retrieve the required information. Then you create new queries and collections using WQL.

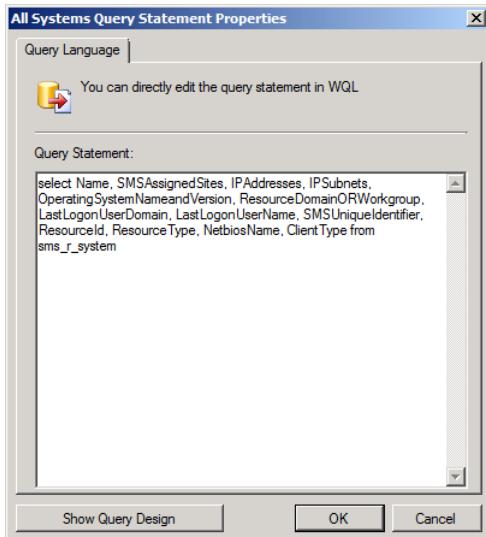
1. In the Configuration Manager console, select the Monitoring workspace, and then select Queries.



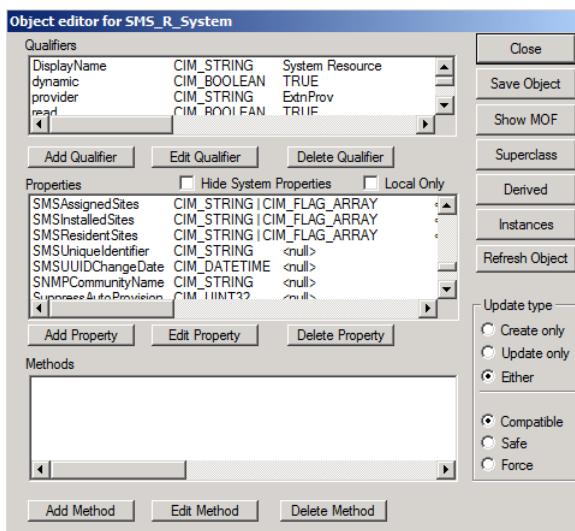
- Right-click the All Systems query, select Properties, Edit Query Statement, and then under the General tab click Show Query Language.



- In the All Systems Query Statement Properties dialog box, you can see that the SMS_R_System Class is queried to retrieve the Configuration Manager object.



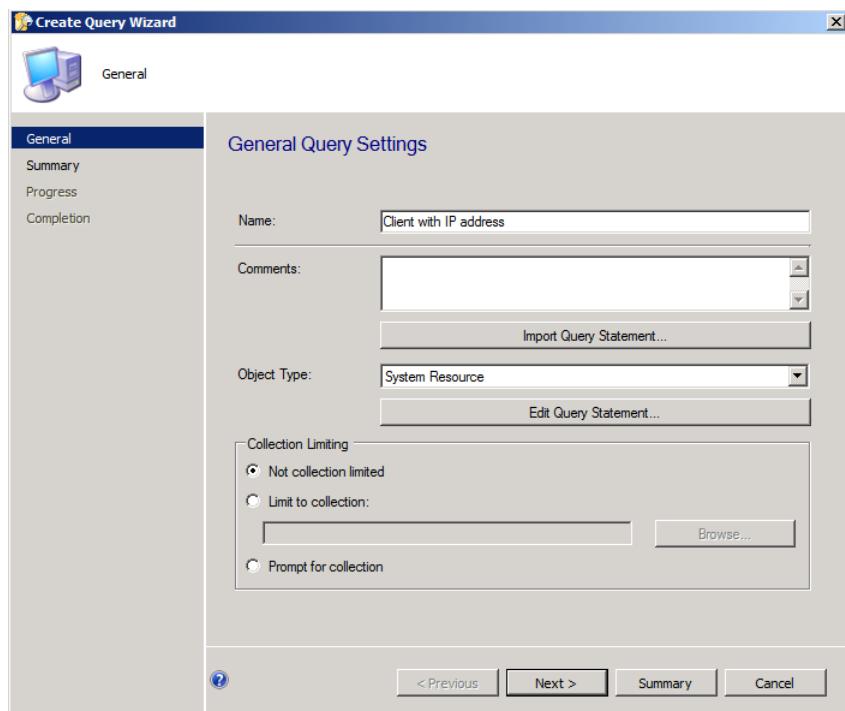
4. Next use Wbemtest to see the different properties of this class used in the query. Launch Wbemtest, connect to the root\SMS\site_<siteCode> namespace, click Enum Classes, select Recursive, click OK, and navigate to the SMS_R_System class. Double-click SMS_R_System to open its properties as shown in the following screenshot.



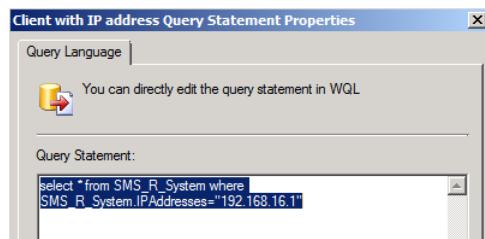
Note that Windows Query Language (WQL) statements and SQL statements often appear similar but are actually quite different in how they are used by Configuration Manager. For example, WQL returns data by querying WMI classes while SQL returns data by querying the database. And while WQL is used in Configuration Manager queries and collections, SQL is used for custom reporting. So when you need to deal with WQL, you can use the Wbemtest tool, and when you need to work with SQL

statements you can use SQL Management Studio. So as you can see, WQL and SQL are two different mechanisms, one for querying data from the SMS provider and the other from the SQL database. To continue with this example, now use the SMSprov.log to see how WQL interacts with the SMS provider, which converts WQL Statements into their equivalent SQL statement.

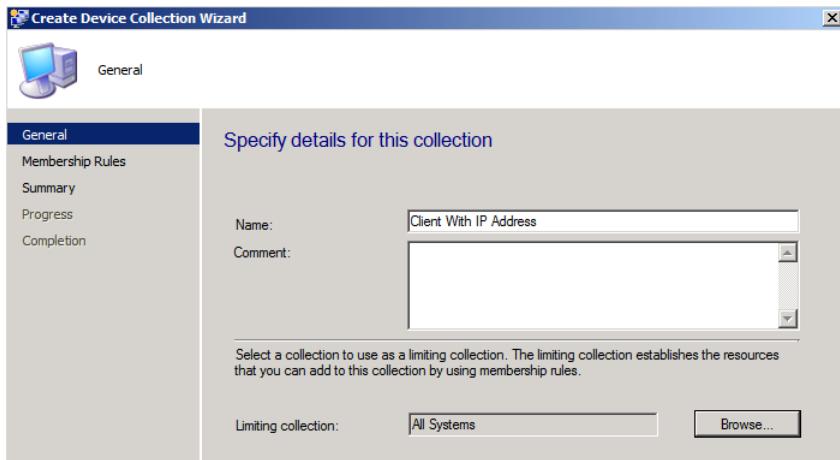
5. Select the Monitoring workspace, then select Queries, then create a new query as shown in the following screenshot:



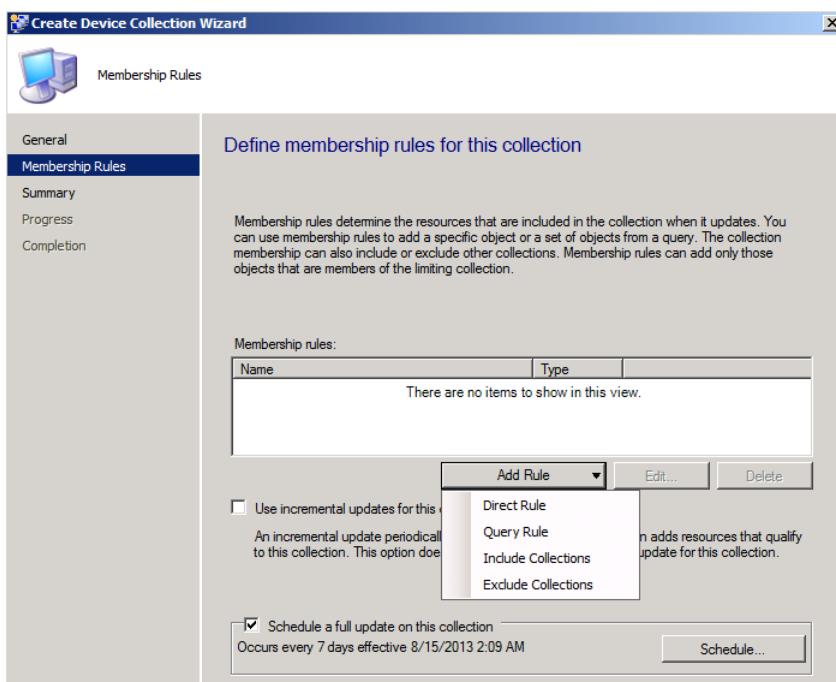
6. On the General page, click Edit Query Statement, click Show Query Language, and then type the WQL query as shown in the following screenshot.



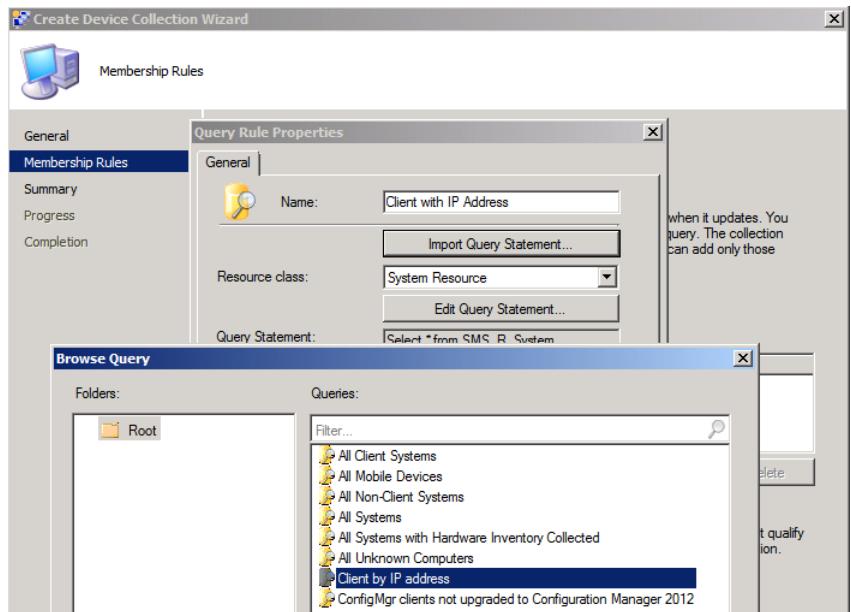
7. Select the Assets And Compliance workspace, select Device Collections, then create a device collection as shown below:



8. On the Membership Rules page, click Add Rule, and select Query Rule.



9. In the Query Rule Properties dialog box, click Import Query Statement. Then in the Browse Query dialog box under Queries, select the query created previously in step 7.



10. Click Next on all subsequent wizard pages, and then right-click the collection to update the collection membership.
11. Open the SMSProv.log and look for the WQL statement pertaining to the query you created. The log file entry selected in the following screenshot shows the WQL statement from the example in this procedure.

Configuration Manager Trace Log Tool - [C:\Program Files\Microsoft Configuration Manager\Logs\SMSProv.log]			
Log Text	Component	Date/Time	Thread
ExecQueryAsync: START select * from SMS_R_System where SMS_R_Syste...	SMS Provider	7/13/2013 8:41:25 AM	3312 (0xCF0)
Adding Handle 26796264 to async call map	SMS Provider	7/13/2013 8:41:25 AM	3312 (0xCF0)
CExProviderClassObject::DoCreateInstanceEnumAsync (SMS_Query)	SMS Provider	7/13/2013 8:41:25 AM	3312 (0xCF0)
CSpQueryForObject :: Execute...	SMS Provider	7/13/2013 8:41:25 AM	3312 (0xCF0)
CExUserContext::LeaveThread : Releasing IWbemContextPtr=56998656	SMS Provider	7/13/2013 8:41:25 AM	3524 (0xDC4)
Execute WQL =select * from SMS_R_System where SMS_R_System.IPA... 7/13/2013 8:41:25 AM	SMS Provider	7/13/2013 8:41:25 AM	3312 (0xCF0)
Execute SQL =select distinct SMS_R_System.ItemKey,SMS_R_System.Activ...	SMS Provider	7/13/2013 8:41:25 AM	3312 (0xCF0)
Results returned : 1 of 2	SMS Provider	7/13/2013 8:41:26 AM	3312 (0xCF0)
Removing Handle 26796264 from async call map	SMS Provider	7/13/2013 8:41:26 AM	3312 (0xCF0)
ExecQueryAsync: COMPLETE select * from SMS_R_System where SMS_R_Syste...	SMS Provider	7/13/2013 8:41:26 AM	3312 (0xCF0)
CExUserContext::LeaveThread : Releasing IWbemContextPtr=56998880	SMS Provider	7/13/2013 8:41:26 AM	3312 (0xCF0)
CExUserContext::EnterThread : User =CORP\kaushalp Sid=0x0105000000...	SMS Provider	7/13/2013 8:41:40 AM	3312 (0xCF0)
Context: SMSAppName=Configuration Manager Administrator console	SMS Provider	7/13/2013 8:41:40 AM	3312 (0xCF0)
Date/Time: 7/13/2013 8:41:25 AM Component: SMS Provider			
Thread: 3312 (0xCF0) Source:			
Execute WQL =select * from SMS_R_System where SMS_R_System.IPA... 7/13/2013 8:41:25 AM			
Elapsed time is 5h 15m 46s 606ms (18946.606 seconds)			

- 12.** The line just below the WQL statement shows the equivalent SQL statement.

The screenshot shows the Configuration Manager Trace Log Tool window. The title bar reads "Configuration Manager Trace Log Tool - [C:\Program Files\Microsoft Configuration Manager\Logs\SMSProv.log]". The menu bar includes File, Tools, Window, Help. Below the menu is a toolbar with icons for Log, Text, XML, Binary, and Hex. A table displays log entries with columns: Log Text, Component, Date/Time, and Thread. One entry is highlighted:

```
Execute SQL =select distinct SMS_R_System.ItemKey,SMS_R_System.Active0,SMS_R_System.AD_Site_Name0,SMS_R_System.AlwaysInternet0,SMS_R_System.AMTFullVersion0,SMS_R_System.ATMStatus0,SMS_R_System.Client0,SMS_R_System.AgentEdition0,SMS_R_System.Client_Type0,SMS_R_System.Client_Version0,SMS_R_System.CPUType0,SMS_R_System.Creation_Date0,SMS_R_System.Decommissioned0,SMS_R_System.Distinguished_Name0,SMS_R_System.EAS_DeviceID,SMS_R_System.Hardware_ID0,SMS_R_System.InternetEnabled0,SMS_R_System.Is_AOAC_Capable0,SMS_R_System.Is_MachineChanges_Persisted0,SMS_R_System.IsClientAMT30Compatible0,SMS_R_System.Is_Assigned_To_User0,SMS_R_System.Is_Portable_Operating_System0,SMS_R_System.Is_Virtual_Machine0,SMS_R_System.Is_Write_Filter_Capable0,SMS_R_System.Last_Logon_Timestamp0,SMS_R_System.User_Domain0,SMS_R_System.User_Name0,SMS_R_System.Name0,SMS_R_System.Netbios_Name0,SMS_R_System.Object_GUID0,SMS_R_System.Obsolete0,SMS_R_System.Operating_System_Name_and0,SMS_R_System.Previous_SMS_UID0,SMS_R_System.Primary_Group_ID0,SMS_R_System.PublisherDeviceID,SMS_R_System
```

Below the table, status information is shown:

Date/Time: 7/13/2013 8:41:25 AM Component: SMS Provider
Thread: 3312 (0xCFO)

Elapsed time is 5h 15m 46s 606ms (18946.606 seconds)

NOTE Unlike SQL tables, SQL views always begin with V_XXX. The SMSProv.log references SQL tables instead of SQL views. Best practice is to always use SQL views when querying the SQL database. For more information on SQL views versus tables, see Chapter 2 of this book.

Interaction between the site server and the Configuration Manager client

Let's now examine how interaction between the site server and Configuration Manager client works. Configuration Manager employs a client-server architecture whereby the site server sends the client policies and configurations. The client in turn stores these policies and configurations in the local WMI repository on the client machine.

Configuration Manager client-specific WMI namespaces can be accessed by Wbemtest as explained earlier in this document. The following section focuses on the client hardware inventory actions and examines how the client's different WMI classes are queried to gather the inventory data and send it to the Configuration Manager site server.

Using Resource Explorer on the site server, you can watch as the hardware inventory client agent captures some of the resource information on the Configuration Manager client

and sends it to the site server. Figure 1-3 shows the Resource Explorer view of one of the Configuration Manager clients.



FIGURE 1-3 Configuration Manager Resource Explorer.

As you can see, the Device ID for the Configuration Manager client is C: and the Volume Serial Number is 5897846F. This information about the Logical Disk resides on the Win32_LogicalDisk WMI class. During the inventory cycle, the Hardware inventory agent queries the WMI class to gather this information and sends it to the site server.

Figure 1-4 shows the Configuration Manager client Win32_LogicalDisk WMI class. This is the same client shown in Figure 1-3. As you can see, the information that resides on the Win32_LogicalDisk is the same as collected and shown in Figure 1-3. You can also examine the inventoryagent.log to see which activities are performed when the hardware inventory cycle occurs.

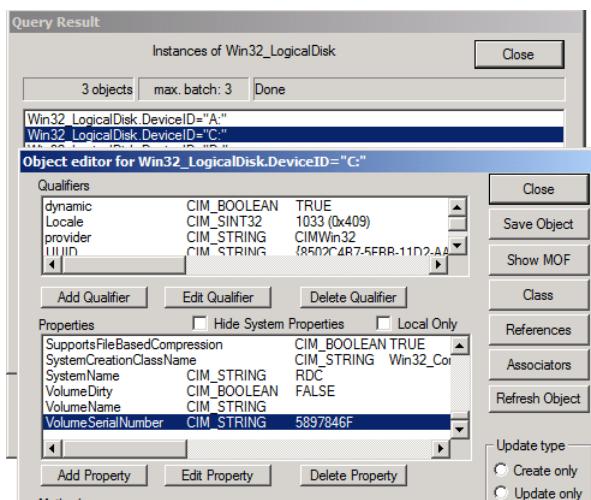


FIGURE 1-4 Win32_LogicalDisk Class

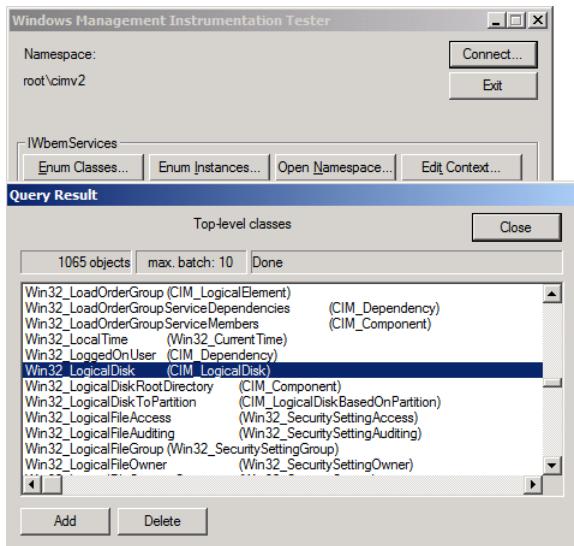


FIGURE 1-4 Win32_LogicalDisk Class (continued)

Reading from the SMS Provider, WMI Scripting, and the Configuration Manager SDK

The SMS Provider is a WMI provider that provides both read and write access to the Configuration Manager site database. The SMS Provider is used by the Configuration Manager console, Resource Explorer, tools, and custom scripts used by Configuration Manager administrators to access site information stored in the site database. The SMS Provider also helps ensure that Configuration Manager object security is enforced by returning only site information that the user account running the console is authorized to view.

The SMS Provider can be installed on the site database server computer, site server computer, or another server-class computer during Configuration Manager setup. After setup has completed, the current installed location of the SMS Provider is displayed on the site properties General tab.

A WMI schema exists to describe the structure of the SMS Provider. Schema namespaces describe the location of Configuration Manager Data within the SMS Provider schema. Table 1-1 contains the common namespaces used by the SMS Provider.

TABLE 1-1 Common namespaces used by the SMS Provider

Namespace	Description
Root\SMS\site_<site code>	The SMS Provider, which is extensively used by the Configuration Manager console, Resource Explorer, Configuration Manager tools, and scripts.
Root\SMS\SMS_ProviderLocation	Provides the location of the SMS provider server for a site.
Root\CIMv2	Location inventoried for WMI namespace information during hardware and software inventory.
Root\CCM	Configuration Manager client configuration policies and client data.
root\CIMv2\SMS	Location of inventory reporting classes that are collected by the inventory client agent. These settings are compiled by clients during computer policy evaluation and are based on the contents of the SMS_def.mof file.

To connect to an SMS provider using VBScript, we need to perform below steps

1. Get a WbemScripting.SWbemLocator object. To set the object reference in visual basic script you can use the following command:

```
Set swbemLocator = CreateObject ("WbemScripting.SWbemLocator")
```

2. Set the authentication level to packet privacy.

```
swbemLocator.Security_.AuthenticationLevel = 6 'Packet Privacy
```

3. Set up a connection to the SMS Provider by using the SWbemLocator object ConnectServer method. Supply credentials only if it is a remote computer.

```
Set net = CreateObject ("WScript.Network")
If UCase (net.ComputerName) = UCase (server) Then
    LocalConnection = true
    userName = ""
    userPassword = ""
    server = "."
End If

' Connect to the server.
Set swbemServices= swbemLocator.ConnectServer (server, root\
sms",userName,userPassword)
If Err.Number<>0 Then
    Wscript.Echo "Couldn't connect: " + Err. Description
    Connect = null
    Exit Function
End If
```

4. Using the SMS_ProviderLocation object ProviderForLocalsite property, connect to the SMS Provider for the local computer and receive a SWbemServices object.

```
' Determine where the provider is and connect.
```

```

Set providerLoc = swbemServices.InstancesOf("SMS_ProviderLocation")

For Each location In providerLoc
    If location.ProviderForLocalsite = True Then
        Set swbemServices = swbemLocator.ConnectServer _
            (location.Machine, "root\sms\site_" + _
                location.SiteCode,userName,userPassword)
    If Err.Number<>0 Then
        Wscript.Echo "Couldn't connect:" + Err.Description
        Connect = Null
        Exit Function
    End If
    Set Connect = swbemServices
    Exit Function
End If
Next
Set Connect = null ' Failed to connect.

```

5. Use the SWbemServices object to access provider objects.

```
On Error Resume Next
```

```

Dim net
Dim localConnection
Dim swbemLocator
Dim swbemServices
Dim providerLoc
Dim location

Set swbemLocator = CreateObject ("WbemScripting.SWbemLocator")
swbemLocator.Security_.AuthenticationLevel = 6 'Packet Privacy.
    ' If the server is local, don't supply credentials.
Set net = CreateObject ("WScript.Network")
If UCase (net.ComputerName) = UCase (server) Then
    LocalConnection = true
    userName = ""
    userPassword = ""
    server = "."
End If

' Connect to the server.
Set swbemServices= swbemLocator.ConnectServer (server, root\
sms",userName,userPassword)
If Err.Number<>0 Then
    Wscript.Echo "Couldn't connect: " + Err. Description
    Connect = null

```

```

        Exit Function
    End If

    'Determine where the provider is and connect.
    Set providerLoc = swbemServices.InstancesOf("SMS_ProviderLocation")

    For Each location In providerLoc
        If location.ProviderForLocalSite = True Then
            Set swbemServices = swbemLocator.ConnectServer _
                (location.Machine, "root\sms\site_" + _
                 location.SiteCode,userName,userPassword)
            If Err.Number<>0 Then
                Wscript.Echo "Couldn't connect:" + Err.Description
                Connect = Null
                Exit Function
            End If
            Set Connect = swbemServices
            Exit Function
        End If
    Next
    Set Connect = null ' Failed to connect.

```

The Configuration Manager SDK, which can be downloaded from <http://msdn.microsoft.com/en-us/library/hh948960.aspx>, contains documentation and samples that are necessary to write applications to access and modify Configuration Manager data. It provides information about extending the Configuration Manager console, and also provides comprehensive reference material for each Configuration Manager feature.

The Configuration Manager SDK provides information applicable to administrators who want to automate Configuration Manager through script and useful to developers adding features and extensions to base Configuration Manager functionality.

CHAPTER 2

Configuration Manager custom reporting

Microsoft System Center 2012 Configuration Manager includes a number of different built-in reports that can be used to retrieve information from the Configuration Manager database. These reports can provide great visibility into your overall environment. Configuration Manager uses Microsoft SQL Server as its backend database engine. During hardware and software scans, Configuration Manager scans the local machine and sends the information collected to the backend database to be stored there. Since the data collected resides in an SQL database, Transact-SQL (T-SQL) queries can be used to query the database and retrieve information about the data stored in it.

Configuration Manager creates several database tables and views during the site server installation. These views and tables are queried by built-in queries to generate the reports. If the default reports are not sufficient for your needs, however, you can generate custom reports by creating custom T-SQL queries. This chapter discusses design issues concerning the SQL database and underlying tables and views used for querying the database and generating custom reports.

Configuration Manager SQL database design

During the installation of the Configuration Manager 2012, the Configuration Manager database is created. By default the name of the Configuration Manager database is CM_<SiteCode>. The Configuration Manager database is spread across multiple tables.

To examine the database using SQL Management Studio, go to All Programs, Microsoft SQL Server, SQL Server Management Studio, and select CM_<SiteCode> to open SQL Management Studio. If you expand the CM_<SiteCode> database, you will see all the database objects pertaining to the Configuration Manager database as shown in Figure 2-1.

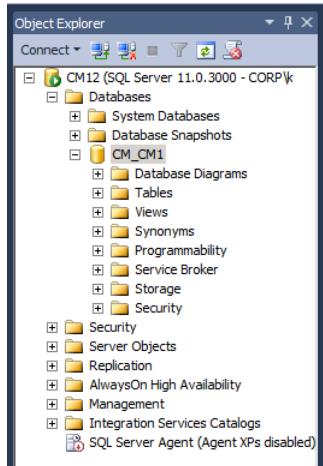


FIGURE 2-1 The Configuration Manager database in SQL Management Studio.

Configuration Manager database tables and views

Configuration Manager creates several tables and views during the installation of the site server. Having a solid understanding of these tables and views is critical for creating custom reports in Configuration Manager. This section provides some tips for identifying the relevant table or view to be used for custom SQL queries.

Open SQL Management Studio, select New Query, and then select Configuration Database. Next run the following query to list all configuration database tables:

```
Select * From Information_Schema.Tables order by Table_Name
```

The query result is shown in Figure 2-2.

The screenshot shows a SQL Server Management Studio window with a query editor and a results grid. The query in the editor is:

```
Select * From INFORMATION_SCHEMA.TABLES
Order by TABLE_NAME
```

The results grid displays 123 rows of data with the following columns:

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
112	CM_CM1	dbo	BGB_ResStatus	BASE TABLE
113	CM_CM1	dbo	BGB_ResStatusHistory	BASE TABLE
114	CM_CM1	dbo	BGB_ResStatusPending	BASE TABLE
115	CM_CM1	dbo	BGB_ResTask	BASE TABLE
116	CM_CM1	dbo	BGB_ResTaskHistory	BASE TABLE
117	CM_CM1	dbo	BGB_ResTaskPush	BASE TABLE
118	CM_CM1	dbo	BGB_ResTaskPushHistory	BASE TABLE
119	CM_CM1	dbo	BGB_ResTaskPushPending	BASE TABLE
120	CM_CM1	dbo	BGB_Server	BASE TABLE
121	CM_CM1	dbo	BGB_ServerHistory	BASE TABLE
122	CM_CM1	dbo	BGB_Task	BASE TABLE
123	CM_CM1	dbo	BGB_TaskTemplate	BASE TABLE

At the bottom of the results grid, a message states: "Query executed successfully." Below the grid, status information includes: CM12 (11.0 SP1) | CORP\kaushalp (68) | CM_CM1 | 00:00:00 | 2562 rows.

FIGURE 2-2 Configuration Manager tables and views.

You can search any tables or columns in the Configuration Manager database. For example, you might want information about MAC addresses. Start by finding the table or view that contains this information. To do this, run the following query to get all tables and views that contain a column name like %MacAddress%:

```
Select * from INFORMATION_SCHEMA.COLUMNS
Where COLUMN_NAME like '%MacAddress%'
```

The result is shown in Figure 2-3.

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME
1	CM_CM1	dbo	v_GS_NETWORK_ADAPTER_CONFIGURATION	MACAddress0
2	CM_CM1	dbo	v_HS_NETWORK_ADAPTER_CONFIGURATION	MACAddress0
3	CM_CM1	SCCM_Ext	vex_GS_NETWORK_ADAPTER_CONFIGURATION	MACAddress0
4	CM_CM1	dbo	UAT_Machine	MacAddress
5	CM_CM1	dbo	Network_DATA	MACAddress00
6	CM_CM1	dbo	Network_HIST	MACAddress00
7	CM_CM1	dbo	PendingRegistrationData	MACAddress
8	CM_CM1	dbo	v_Network_DATA_Serialized	MACAddress0
9	CM_CM1	dbo	vSMS_PendingRegistrationData	MACAddress
10	CM_CM1	dbo	v_GS_NETWORK_ADAPTER	MACAddress0
11	CM_CM1	dbo	v_HS_NETWORK_ADAPTER	MACAddress0
12	CM_CM1	SCCM_Ext	vex_GS_NETWORK_ADAPTER	MACAddress0
13	CM_CM1	dbo	vMacAddresses	MacAddresses
14	CM_CM1	dbo	v_StateMigration	RestoreMACAddress

FIGURE 2-3 Searching the database for MAC addresses.

The difference between SQL tables and views

Both views and tables are database object types. In simple terms, *views* are stored or named select queries. *Tables* are made up of columns and rows. A *column* is a set of data, which belongs to a same data type. A *row* is a sequence of values, which can be from different data types. Each column is identified by a column name, and each row is uniquely identified by the table primary key.

Views

The body of each view is a SELECT statement. Views are called the “virtual tables” of the database. Though the views are stored in the database, they are not run until they are called using another SELECT statement. When they are called using SELECT statements, their stored SELECT queries are executed and show the results. Since views have only SELECT queries as their bodies, they do not need a large space. Views provide several benefits.

- Once the view is created, it can be called again and again using its name, without re-writing the SELECT query.
- Since views are pre-compiled objects, they execute faster than executing their SELECT queries (the Body of the views) separately.
- Views can be used to restrict the table data access. Therefore, they can play an important role in data security, as well.

Tables

A table is a collection of rows that store data. Each row in a table can include data from a different data type, and each row must be identified by a unique identifier (primary key). You use INSERT, UPDATE, and DELETE queries to insert a new row, update an existing row value, and delete a row from the table. SELECT queries are used to retrieve data from tables. If necessary, you can use ALTER TABLE queries to change the table structure after it is created. Tables need more space than views to store data content. There are several types of tables in databases.

- Internal tables
- External tables
- Temporary tables

Because views offer many advantages over tables as detailed earlier, it is best to query views instead of tables when you query the Configuration Manager database. In short, when you are querying the Configuration Manager database, always query the configuration views instead of the table.

Common Configuration Manager views

The Configuration Manager database contains several views. It's worthwhile to know some of the common views that contain useful information.

One of the most useful views is the V_R_System view shown in Figure 2-4. This view is derived from the discovery data table. It contains information about the Configuration Manager clients.

TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION
v_R_System	ResourceID	1
v_R_System	ResourceType	2
v_R_System	Active0	3
v_R_System	AD_Site_Name0	4
v_R_System	AlwaysInternet0	5
v_R_System	AMTFullVersion0	6
v_R_System	AMTStatus0	7
v_R_System	Client0	8
v_R_System	AgentEdition0	9
v_R_System	Client_Type0	10

FIGURE 2-4 Configuration Manager view V_R_System

Another frequently used view is V_GS_Workstation_Status, which you can use to pull the last hardware scan time.

V_GS_ADD_REMOVE_PROGRAMS is another useful view that contains add and remove data collected from Configuration Manager clients.

For network adapters, it is common to query the V_GS_NETWORK_ADAPTER view to see the different columns. If the network adapter configuration data is required, start by querying V_GS_NETWORK_ADAPTER_CONFIGUR.

Similarly, you can use the V_GS_COMPUTER_SYSTEM view for determining the hardware model and manufacturer.

You should develop good familiarity with the different views as you start creating custom queries. But you always need to remember to use views instead of tables when you are querying the database and creating your custom T-SQL queries.

NOTE The configuration database contains different types of views. For example, views that start with V_GS contain current data while those that start with V_HS contain historical data.

Working with SQL inner and left outer joins

The Configuration Manager database uses a relational database design, but the information stored in it resides on different database tables in many different places. Writing custom queries not only requires a good understanding of the database table schema but also knowledge of T-SQL. When the information you want resides in a single view, simple SELECT statements work well. However, when the information to be retrieved resides in multiple tables, you must join the tables to get the required data.

For example, if you need to pull the information about all of the machines in the Configuration Manager database, you would use a join. To do this, you run the following T-SQL query:

```
Select Name0 from v_R_System
```

The result is shown in Figure 2-5.

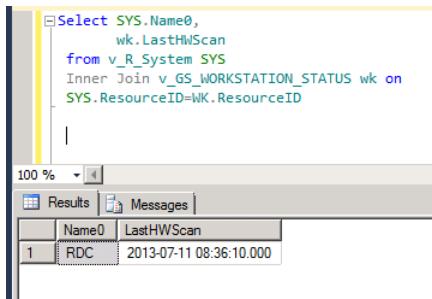
Name0
1 RDC
2 CM12
3 WINXP
4 SQL2012

FIGURE 2-5 Pulling the names of machines in the Configuration Manager database.

Now you might want to add another column that shows when the last hardware inventory occurred for these machines. To do this, you need to reference another view named V_GS_Workstation_Status. You also need to use a join, so start by using the following inner join.

```
Select SYS.Name0,
       wk.LastHwScan
  From v_R_System SYS
Inner Join v_GS_WORKSTATION_STATUS wk on
SYS.ResourceID=WK.ResourceID
```

The results are shown in Figure 2-6.



A screenshot of SQL Server Management Studio (SSMS) showing a query window and a results grid. The query window contains the following T-SQL code:

```
Select SYS.Name0,
       wk.LastHwScan
  From v_R_System SYS
Inner Join v_GS_WORKSTATION_STATUS wk on
SYS.ResourceID=WK.ResourceID
```

The results grid shows one row with the following data:

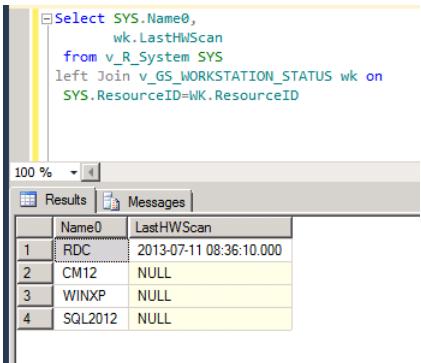
	Name0	LastHwScan
1	RDC	2013-07-11 08:36:10.000

FIGURE 2-6 An example of an inner join and the corresponding result.

As you can see from Figure 2-6, the record set dropped from three records to one. An inner join returns the matching records in both tables. The V_GS_Workstation_Status view contains only the record for the one machine shown in this figure, while the V_R_System view contains the three records shown previously in Figure 2-5. So an inner join by design looks for matching records in both tables.

But what if you need to report the data for all machines irrespective of whether the LastHwScan data for the machine is available? In this scenario you could use a left outer join (commonly known as a *left join*) as shown in Figure 2-7. Despite the fact that some of the data is not available in the V_GS_Workstation_Status table, the SQL query returned all the data set available on the V_R_SYSTEM view. Here is the query you use:

```
Select SYS.Name0
       wk.LastHwScan
  From v_R_System SYS
Left Join v_GS_WORKSTATION_STATUS wk on
SYS.ResourceID=WK.ResourceID
```



The screenshot shows a SQL query window with the following code:

```
Select SYS.Name0,
       wk.LastHWScan
  from v_R_System SYS
 left Join v_GS_WORKSTATION_STATUS wk on
      SYS.ResourceID=wk.ResourceID
```

The results pane displays a table with four rows:

	Name0	LastHWScan
1	RDC	2013-07-11 08:36:10.000
2	CM12	NULL
3	WINXP	NULL
4	SQL2012	NULL

FIGURE 2-7 An example of an outer join and the corresponding result.

These examples illustrate the power of using inner and outer joins when querying the Configuration Manager database.

PART II

System Center 2012 Configuration Manager reporting

Reporting is a powerful tool within Configuration Manager. It provides an inside view of your infrastructure's health. For those of you familiar with legacy Reporting Point in Configuration Manager 2007, SQL Server Reporting Service (SSRS) is completely different and provides much more flexibility than legacy Reporting Point. SSRS is a server-based reporting tool that provides reporting functionality for a variety of data sources. The three chapters in Part II are designed to familiarize you with SSRS and show you how to integrate SSRS with Microsoft System Center 2012 Configuration Manager and to customize and create new reports using SSRS.

This page left intentionally blank.

Integrating SQL Server Reporting Services with Configuration Manager 2012

In Microsoft System Center 2012 Configuration Manager the legacy reporting point has been replaced with the Reporting services point, which uses SQL Server Reporting Services (SSRS). A common question is: How can you use and customize reports using SSRS? If you aren't familiar with SQL queries, you might be hesitant to use SSRS. This chapter shows you how easy it is to create custom reports. However, before you start using SSRS, the first step is to integrate SSRS with Configuration Manager.

This chapter covers the process of installing an SSRS services point using the Configuration Manager console. The next chapter describes the process of customizing and running reports using SSRS.

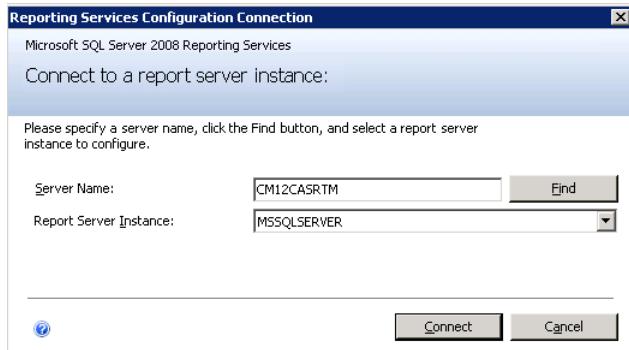
Prerequisites for reporting services points in Configuration Manager 2012

There are some prerequisites before you are able to integrate SSRS with Configuration Manager 2012. Before you can install a reporting services point in Configuration Manager, you must ensure that SSRS is installed and configured properly. The following example uses Microsoft SQL Server 2008.

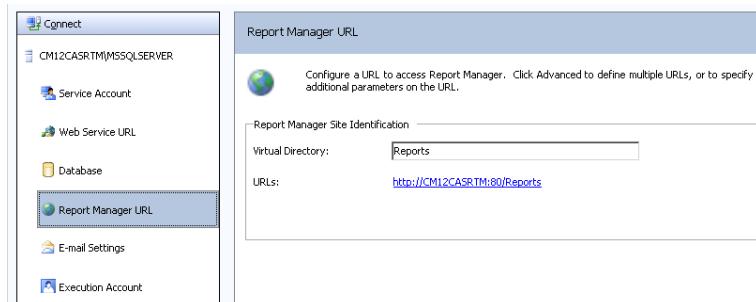
1. Click Start, select All Programs, Microsoft SQL Server 2008, Configuration Tools, and then select Reporting Services Configuration Manager.



2. In the Reporting Services Configuration Connection dialog box, click Connect.



3. On the Report Server Status page, make sure Report Service Status is set to Started. If not, click Start to start the service.
4. On the Web Service URL page, click Apply to configure the Report Server Web Service. This creates a virtual directory called ReportServer.
5. Click the Database tab and make sure Report Server Mode is set to Native. If you are setting this for the first time, you are prompted to create a new report server database or use an existing one. Choose the Create A New Report Server Database option and then click Next.
6. Click Test Connection to make sure it works.
7. Specify the Database Name (the default is ReportServer), set the Report Server Mode to Native, and click Next.
8. Click Next on the Credentials page to accept the default settings.
9. Click Next on the Summary page to complete the wizard.
10. On the Report Manager URL page, click Apply to setup the Report Manager virtual directory. This creates a virtual directory called Reports.
11. On the Report Manager URL page, click the URLs link for *http://<Servername>:80/Reports* to make sure you are able to connect. It might ask you for a username and password.



Installing a Reporting services point

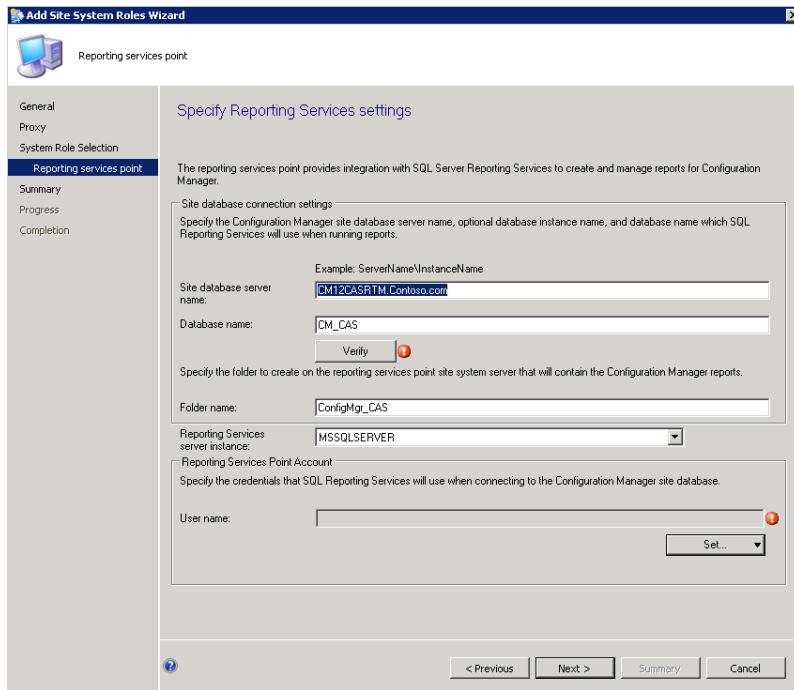
After you confirm the installation and configuration of SSRS, the next step is to integrate SSRS with Configuration Manager 2012 by installing a Reporting services point using the Configuration Manager console. This process copies the report folders and reports to SSRS and applies the appropriate security for the folders and reports. After you install the reporting services point, do not change the SSRS Report Server URL because Configuration Manager will continue to use the original URL and you will be unable to run the reports. To change the URL, you must uninstall the reporting services point, change the URL, and then reinstall the reporting services point.

To install a reporting services point, follow these steps:

1. Connect to the Configuration Manager console.
2. Select Administration, Site Configuration, Server And Site System Roles.
3. Right-click the server on which you want to install the reporting services point and click Add Site System Roles to start the wizard.
4. On the General page, accept the default settings and click Next.
5. On the Proxy page, accept the default settings and click Next.
6. On the System Role Selection page, select Reporting Services Point and click Next.
7. On the Reporting Services Point page, specify the Configuration Manager site database server name and the database name.

IMPORTANT If your SQL server is remote, specify the SQL server name and not the Configuration Manager database server name. If you accidentally specify the Configuration Manager database server name when your SQL server is remote, you might see the following error message: “Unable to locate any configured SRS instances on the server. Verify SRS is installed, accessible, and correctly configured.”

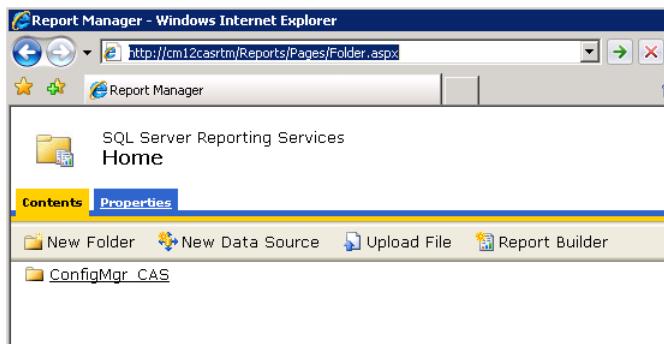
8. Click Verify to make sure the wizard has access to the site database.
9. Specify the folder name that will host the Configuration Manager reports.
10. Click Set to specify a reporting services point account and click Next. The reporting services point account must have Read rights to the site database.



- 11.** On the Summary page, click Next to complete the Reporting Services Point Installation Wizard.
- 12.** On the Completion page, click Close.
- 13.** Review srsrpMSI.log and srsrpsetup.log for any errors. These logs are located in the %ProgramFiles%\Microsoft Configuration Manager\Logs directory.

```
Log Text
<08/12/13 12:36:23> No versions of SMSRSRP are installed. Installing new SMSRSRP.
<08/12/13 12:36:23> Enabling MSI logging. srsrp.msi will log to C:\Program Files\Microsoft Configuration Manager\logs\srsrpMSI.log
<08/12/13 12:36:23> Installing C:\Program Files\Microsoft Configuration Manager\bin\x64\srsrp.msi SRSPINSTALLDIR="C:\Program Files\SMS_SRSRP" SRSRPLANGPACKFLAGS=0
<08/12/13 12:37:49> srsrp.msi exited with return code: 0
<08/12/13 12:37:49> Installation was successful.
<08/12/13 12:37:49> CTool:RegisterManagedBinary: run command line: "C:\Windows\Microsoft.NET\Framework64\v4.0.30319\RegAsm.exe" "C:\Program Files\SMS_SRSRP\srsrvserver.dll" /unregister
<08/12/13 12:37:50> CTool:RegisterManagedBinary: Registered C:\Program Files\SMS_SRSRP\srsrvserver.dll successfully
<08/12/13 12:37:50> Unregistered DLL C:\Program Files\SMS_SRSRP\srsrvserver.dll
<08/12/13 12:37:50> CTool:RegisterManagedBinary: run command line: "C:\Windows\Microsoft.NET\Framework64\v4.0.30319\RegAsm.exe" "C:\Program Files\SMS_SRSRP\srsrvserver.dll" /codebase
<08/12/13 12:37:55> CTool:RegisterManagedBinary: Registered C:\Program Files\SMS_SRSRP\srsrvserver.dll successfully
<08/12/13 12:37:55> Registered DLL C:\Program Files\SMS_SRSRP\srsrvserver.dll
<08/12/13 12:37:55> No version of SMSRSRP are installed. Installing new SMSRSRP.
<08/12/13 12:37:55> Enabling MSI logging. srsrp.msi will log to C:\Program Files\Microsoft Configuration Manager\logs\srsrpMSI.log
<08/12/13 12:37:55> Installing C:\Program Files\Microsoft Configuration Manager\bin\x64\srsrp.msi SRSPINSTALLDIR="C:\Program Files\SMS_SRSRP" SRSRPLANGPACKFLAGS=0
<08/12/13 12:38:30> srsrp.msi exited with return code: 0
<08/12/13 12:38:30> Installation was successful.
<08/12/13 12:38:30> Installation was successful.
<08/12/13 12:38:30> "RoleSetup"
```

- 14.** When the reporting services point is installed successfully, connect to <http://<Servername>/Reports> to make sure the Configuration Manager database is listed and you are able to access it.



This page left intentionally blank.

Customizing SSRS reports for Configuration Manager 2012

The previous chapter describes the process of setting up SSRS and integrating it with Microsoft System Center 2012 Configuration Manager. This chapter discusses customizing Configuration Manager reports using SSRS.

The following procedure demonstrates how to create a custom SSRS report for Configuration Manager 2012 using Microsoft SQL Server 2008 and Microsoft Visual Studio 2008.

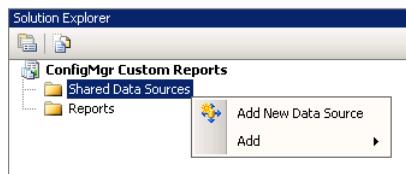
1. Click Start, All Programs, SQL Server 2008, and then select SQL Server Business Intelligence Development Studio. This opens Microsoft Visual Studio 2008.



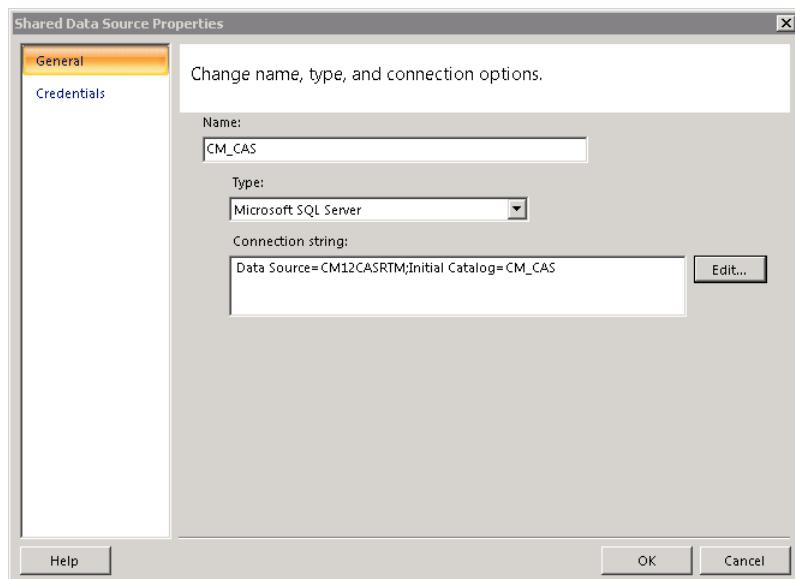
IMPORTANT If SQL Server Business Intelligence Development Studio is not installed, you can install it using the SQL Server CD. Run the setup and select Business Intelligence Development Studio listed under Features on the Feature Selection page.

2. Click File, New, Project, and under Visual Studio Installed Templates choose Report Server Project.
3. Specify the name, location, and solution name (such as, ConfigMgr Custom Reports) and click OK.
4. On the menu bar, click View, Solution Explorer.

5. In Solution Explorer, under ConfigMgr Custom Reports, right-click Shared Data Sources and select Add New Data Source.

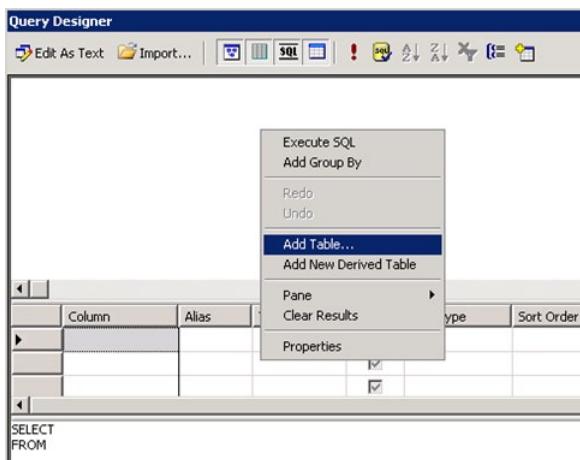


6. On the Shared Data Source Properties page, enter the following information:
 - For Name, specify the name of the Configuration Manager database (such as, CM_CAS).
 - For Type, specify Microsoft SQL Server.
 - For Connection String, click Edit.

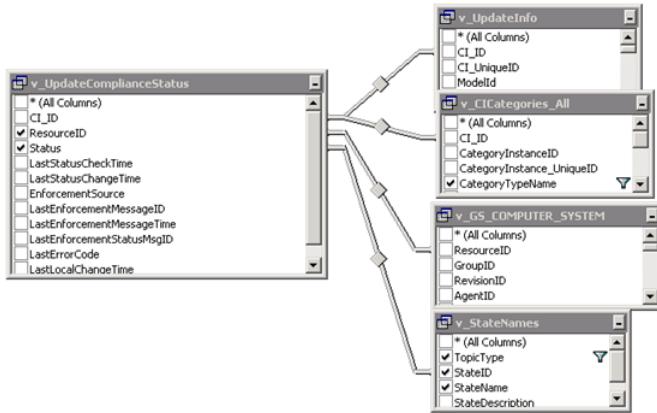


7. Under Connection Properties specify the name of the SQL server (i.e., CM12CASRTM).
8. Select Use Windows Authentication for logging on to the server.
9. Select or enter a Configuration Manager database name (i.e., CM_CAS).
10. Click Test Connection to make sure the connection works.
11. Click OK twice to complete the wizard. This creates an .rds file (i.e., CM_CAS.rds) under Shared Data Sources.
12. In Solution Explorer, right-click Reports and select Add New Report.
13. On the Welcome page click Next.

- 14.** On the Select The Data Source page, select the Configuration Manager database (i.e., CM_CAS) and click Next.
- 15.** On the Design The Query page, click Query Builder to open the Query Designer page.
- 16.** Right-click anywhere on blank portion at the top of the page and select Add Table.



- 17.** Click the View tab and choose the following views:
 - v_UpdateComplianceStatus
 - v_UpdateInfo
 - v_GS_Computer_System
 - v_StateNames
 - v_CICategories_All
 - 18.** Next, to join the appropriate columns with the different views, highlight CI_ID on the v_UpdateComplianceStatus list and drag it onto CI_ID on the v_UpdateInfo view.
 - 19.** Create a join between the following pairs by repeating the procedure from step 18:
 - v_UpdateComplianceStatus.Cl_ID and v_CICategories_All.Cl_ID
 - v_UpdateComplianceStatus.ResourceID and v_GS_COMPUTER_SYSTEM.ResourceID
 - v_UpdateComplianceStatus.Status and v_StateName.StateID
- The result now looks like the following screenshot.



- 20.** On the Query Designer page, under the middle tier box, specify the following filters for the corresponding column names:

- CategoryTypeName = 'UpdateClassification'
- BulletinID = 'MS13-004'
- TopicType = 500

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
Name0		v_GS_COMPUTER_SYSTEM	✓			
ModelName		v_CICategories_All	✓			
CategoryTypeName		v_CICategories_All	✓			= N'UpdateClassification'
BulletinID		v_UpdateInfo	✓			= N'MS13-004'
ArticleID		v_UpdateInfo	✓			
StateName		v_StateNames	✓			
StateID		v_StateNames	✓			
TopicType		v_StateNames	✓			= 500

You can also copy and paste the following SQL query to generate the above joins and filter automatically:

```

SELECT v_UpdateComplianceStatus.ResourceID, v_UpdateComplianceStatus.Status,
v_GS_COMPUTER_SYSTEM.Name0, v_CICategories_All.ModelName, v_CICategories_All.
CategoryTypeName, v_UpdateInfo.BulletinID, v_UpdateInfo.ArticleID, v_StateNames.
StateName, v_StateNames.StateID, v_StateNames.TopicType

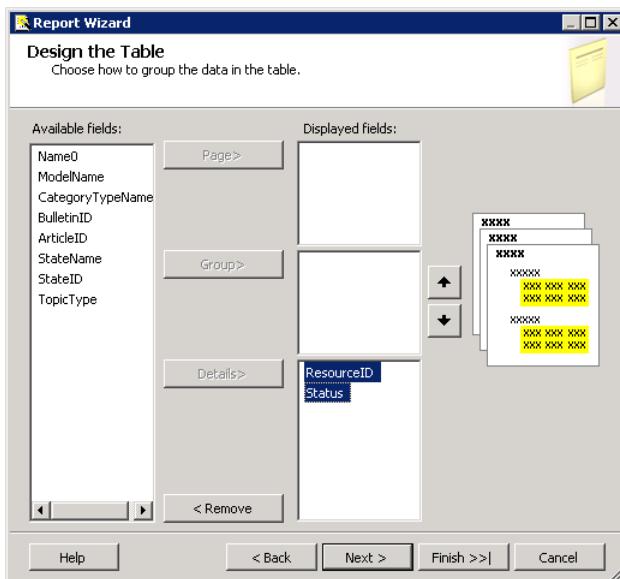
FROM v_UpdateComplianceStatus INNER JOIN v_GS_COMPUTER_SYSTEM ON v_
UpdateComplianceStatus.ResourceID = v_GS_COMPUTER_SYSTEM.ResourceID INNER JOIN
v_CICategories_All ON v_UpdateComplianceStatus.CI_ID = v_CICategories_All.CI_ID
INNER JOIN v_UpdateInfo ON v_UpdateComplianceStatus.CI_ID = v_UpdateInfo.CI_ID
INNER JOIN v_StateNames ON v_UpdateComplianceStatus.Status = v_StateNames.StateID

WHERE (v_CICategories_All.CategoryTypeName = N'UpdateClassification') AND (v_
StateNames.TopicType = 500) AND (v_UpdateInfo.BulletinID = N'MS13-004')

```

- 21.** Click the exclamation mark symbol at the top of the Query Designer page to execute the query and review the results.

22. Click OK to accept the query. This will return you to the Design The Query window.
23. On the Select The Report Type page, select Tabular and click Next
24. On the Design The Table page, select all of the fields under available Fields, click Details to add them to Displayed Fields on the right, and then click Next.



25. On the Choose The Table Style page, select Corporate and click Next.
26. On the Completing The Wizard page, provide a report name, select the Preview Report option at the bottom of the page, and click Finish.
27. This process will create .RDL file, which can be used to upload the report into SSRS.

eBook Custom Report

Resource ID	Status	Name0	Model Name	Category Type Name	Bulletin ID	Article ID	State Name
16777221	2	CM12CASRTM	Site_0D8FA4EB-0030-42E0-A050-34FA20302589/SUM_fba9a3e9-5697-4727-9a5e-df81e68c10e2	UpdateClassification	MS13-004	2742595	Update is required
16777221	2	CM12CA5RTM	Site_0D8FA4EB-0030-42E0-A050-34FA20302589/SUM_c431e423-59a2-4928-a77f-ce62906d8a22	UpdateClassification	MS13-004	2742601	Update is required
16777219	2	CM12PRILA	Site_0D8FA4EB-0030-42E0-A050-34FA20302589/SUM_fba9a3e9-5697-4727-9a5e-df81e68c10e2	UpdateClassification	MS13-004	2742595	Update is required

In the next chapter we'll examine how to customize function-based built-in reports.

This page left intentionally blank.

CHAPTER 5

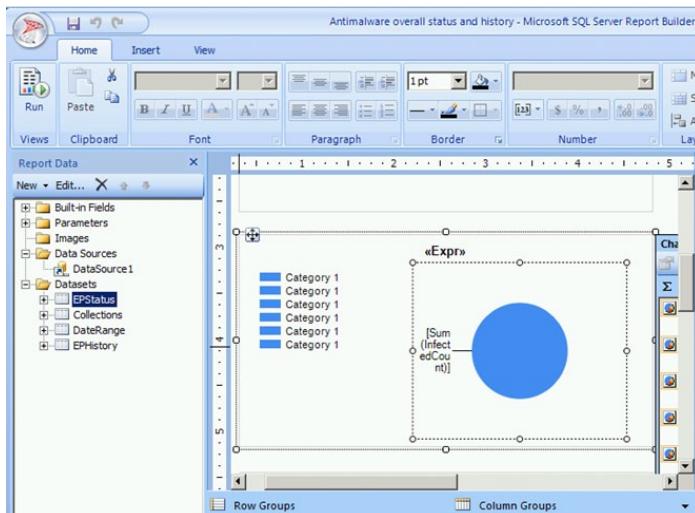
Customizing function-based built-in reports

In Microsoft System Center 2012 Configuration Manager, most of the existing built-in reports are based on function. Many Configuration Manager administrators are reluctant to use SQL, specifically when it comes to programming and function. This chapter attempts to ease those fears with a step-by-step process for customizing function-based built-in reports.

This example uses Endpoint Protection built-in reports and covers the process of customizing and converting function-based into view-based reports. Before starting the procedure, review the query used for the Antimalware Overall Status and History report, which is listed under Endpoint Protection category within Configuration Manager console.

Follow these steps to determine the script being used for the "Antimalware overall status and history" report.

1. Open the Configuration Manager console. Expand Monitoring, Reporting, Reports, Endpoint Protection, right-click Antimalware Overall Status And History report, and then click Edit. The report opens in Report Builder.



2. Under Datasets, right-click each datasets to review the query. Click Database Properties to determine what function is being used to create this report. For example, EPStatus datasets is using the following query:

```
SELECT * FROM fn_rbac_EndpointProtectionHealthStatus(@UserIDs)
Where CollectionID=@CollID
```

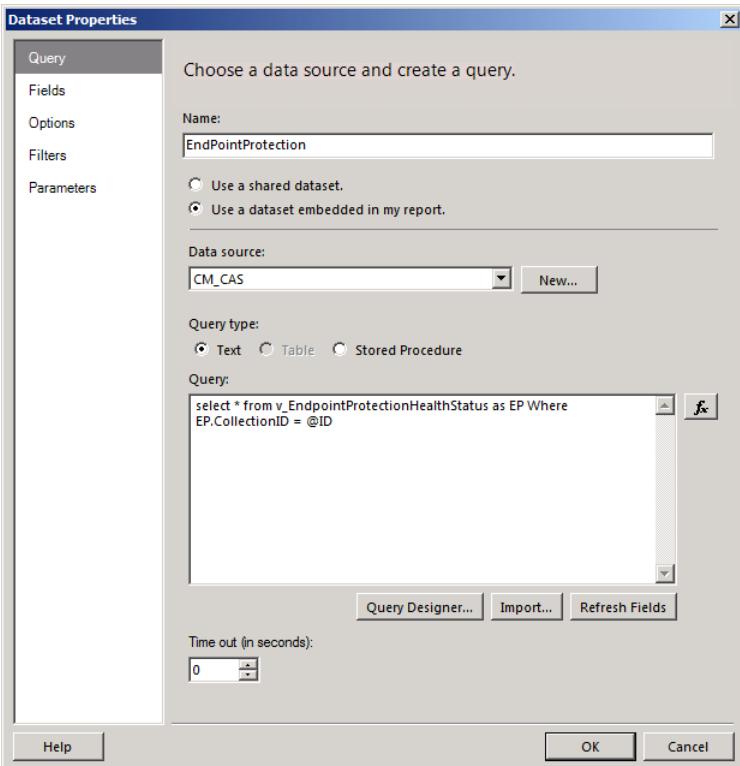
IMPORTANT As you can see, the query has two variables: @UserIDs and @CollID. The UserIDs variable checks RBAC role-based security for existing logged-on users; it verifies the rights for that particular object. CollID allows you to limit any collection for this particular report. The report uses the fn_rbac_EndpointProtectionHealthStatus function.

3. Open SQL Management Studio and expand <ServerName>, Databases, CM_<sitecode>, Programmability, Functions, Table-valued Functions, and then select dbo.fn_rbac_EndpointProtectionHealthStatus.
4. Right-click dbo.fn_rbac_EndpointProtectionHealthStatus and select Script Function As, Create To, and New Query Editor Window to see the code and view that is being used to create the function. In this case, it is using the v_EndpointProtectionHealthStatus view:

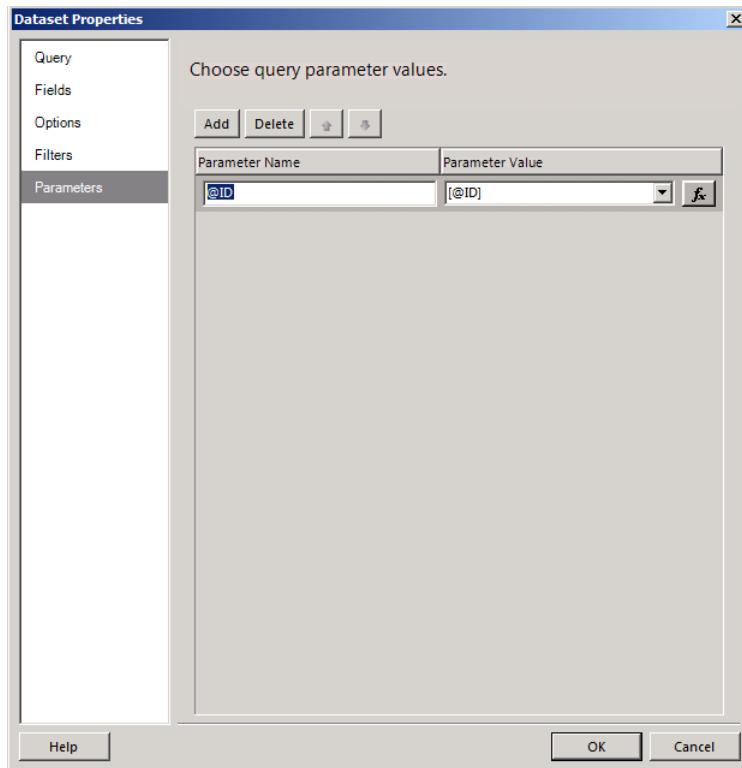
```
CREATE FUNCTION [dbo].[fn_rbac_EndpointProtectionHealthStatus] (@UserIDs nvarchar(max))
RETURNS TABLE
AS
RETURN
(
    SELECT * FROM v_EndpointProtectionHealthStatus
)
```

5. Repeat steps 2 through 4 above for other datasets (as shown below, for example) to determine which views are being used. When you have all of the datasets details, use the following query and options to create shared data sources and reports as described in Chapter 4, "Customizing SSRS reports for Configuration Manager 2012."

```
Select * from v_EndpointProtectionHealthStatus as EP where EP.CollectionID=@ID
```



6. Review the other properties, such as Parameters. Notice that it is auto-populated with Parameter Name and Value.



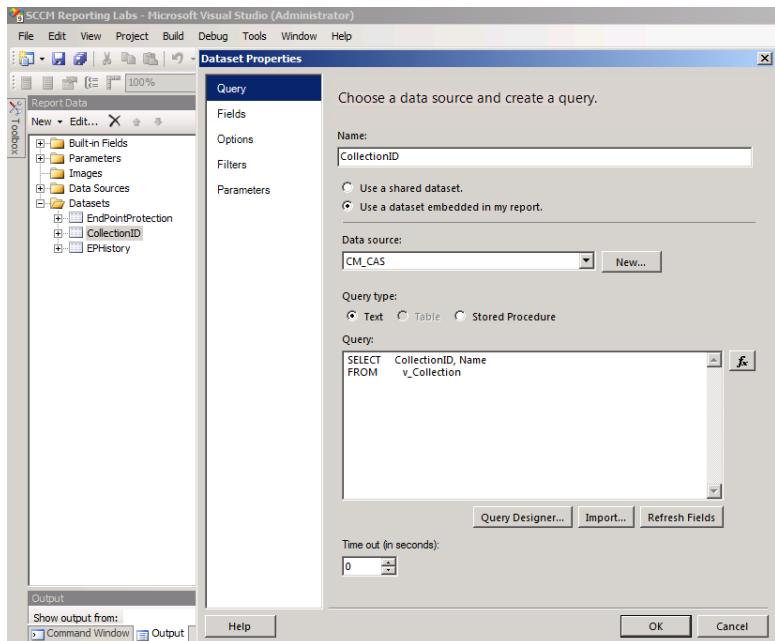
7. Create additional dataset for CollectionID and EPHistory using the following queries:

- CollectionID datasets:

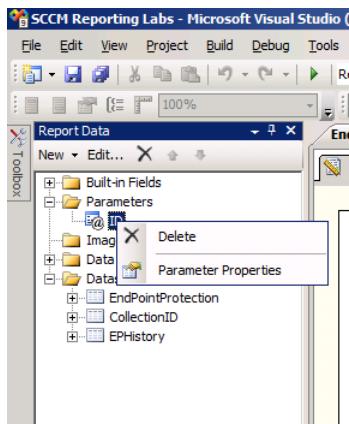
```
Select CollectionID, Name from v_Collection
```

- EPHistory datasets:

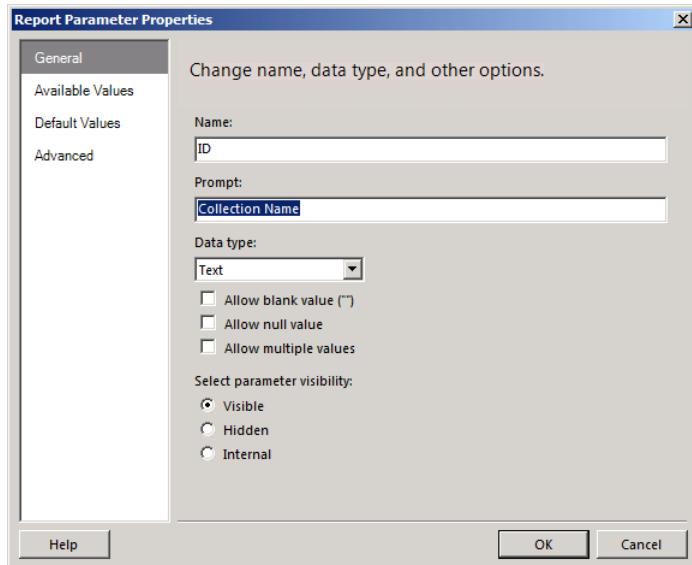
```
Select * from v_EndpointProtectionHealthStatus_History
```



8. Under Parameters, right-click @ID, and select Parameter Properties to open the Report Parameter Properties dialog box.

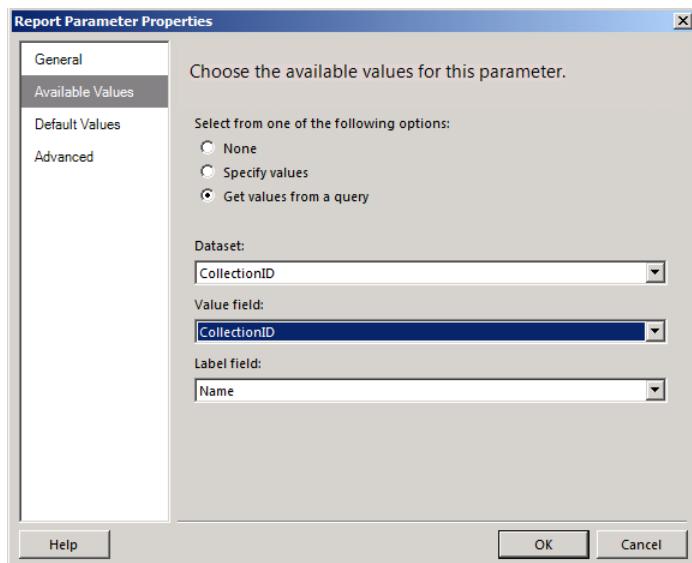


9. On the General tab, specify the Name value as **ID** and the Prompt value as **Collection Name**.



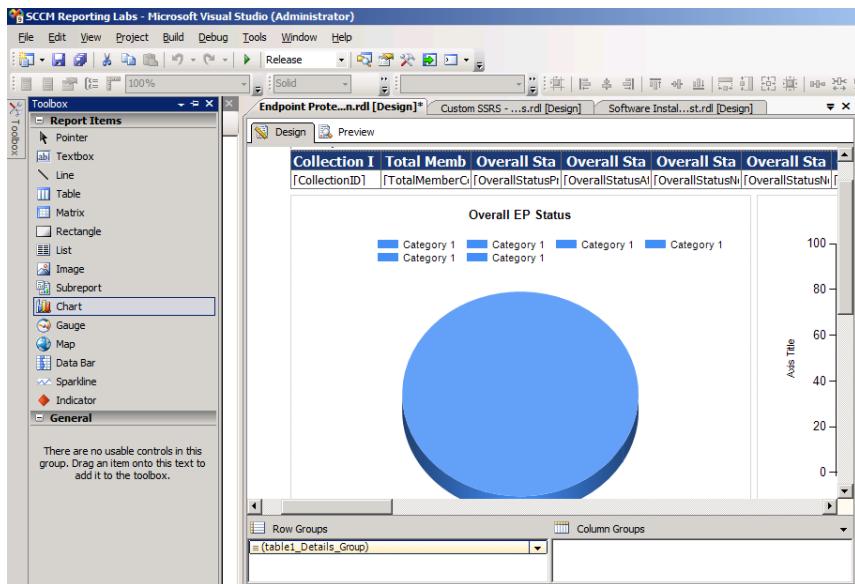
10. On the Available Values tab, select the Get Values From A Query option, specify the following values, and then click OK.

- Dataset: CollectionID
- Value field: CollectionID
- Label field: Name

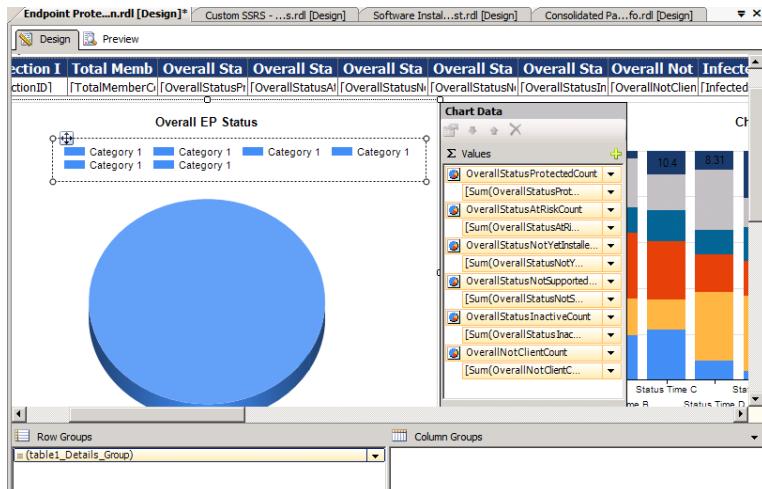


11. On the Default Values and Advanced tabs, leave the default values and click OK to complete the process.

- 12.** Next, add a chart to this report so that you can use it as a dashboard. To do this, go to Toolbox, click Chart, and click anywhere on the Design view. Select Pie Shape for the chart and click OK. The resulting chart will look something like the following figure:



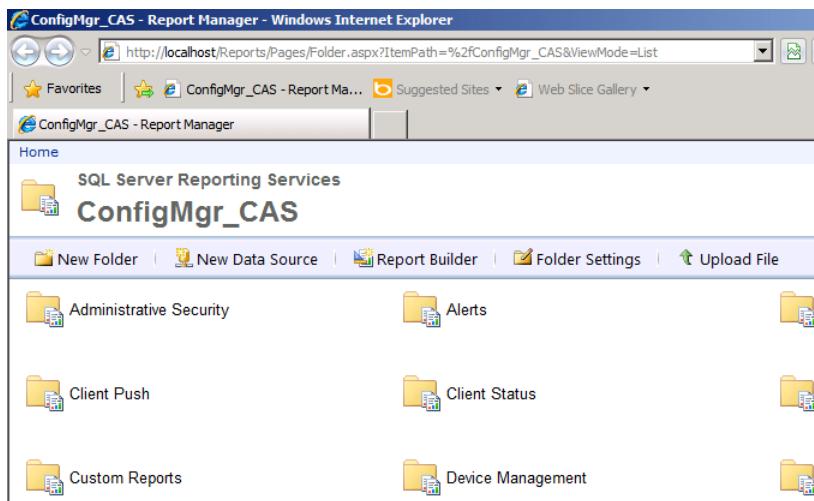
- 13.** Under Overall EP Status in the Design view, select a category box to add different values to be used in the pie chart. The available values appear in the Chart Data pane to the right of the Design view.



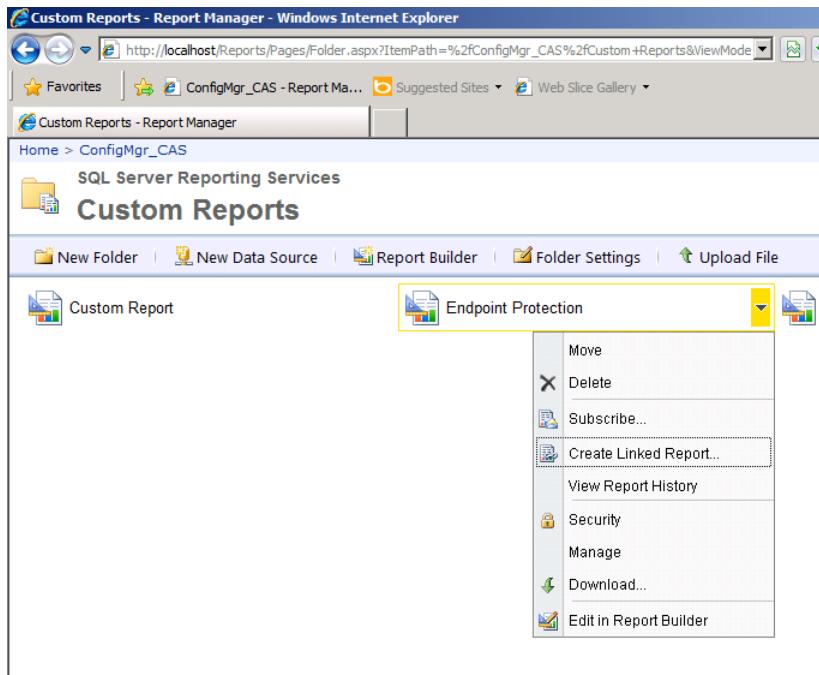
- 14.** Repeat the same process (i.e., steps 12 through 14 above) for different chart types as shown below. Click Preview, select Collection Name from the drop-down menu, and click View Report.



15. When you are satisfied with the results and the view of your report, it is time to upload it to your SSRS site. Connect to your SSRS site by typing **<http://localhost/Reports>** if you are logged on locally on your SSRS site or **<http://<servername>/Reports>**.
16. Click the ConfigMgr_CAS folder to view all the built-in reports for Configuration Manager.
17. Create a folder called CUSTOM REPORTS by clicking the New Folder tab, renaming the folder CUSTOM REPORTS, and then clicking OK.
18. Click the CUSTOM REPORTS folder you just created to select it and then click Upload File. When prompted for the file name, click Browse and go to the folder where you saved the .rdl file for this report, highlight the .rdl file, click Open, and then click OK to upload the file.



- 19.** Hover over the report you just uploaded, click the drop-down menu, and select Manage.



IMPORTANT If you are not running SQL Server 2008 R2 version, you might not get an option to Manage directly from the page. If you just try to run the report without actually setting up the proper data sources, you will get the following error: "The report server cannot process the report or shared dataset. The shared data source 'CM_CAS' for the report server or SharePoint site server is not valid. Browse to the server or site and select a shared data source. (rsInvalidDataSourceReference)."

- 20.** Click Data Sources and make sure A Shared Data Source is selected. Click Browse and expand the ConfigMgr_CAS folder.

The screenshot shows the 'Endpoint Protection - Report Manager' interface in a browser window. The left sidebar has a 'Data Sources' section selected. The main area shows a tree view of report categories under 'ConfigMgr_CAS', including 'Administrative Security', 'Alerts', 'Asset Intelligence', 'Client Push', 'Client Status', 'Compliance and Settings Management', 'Custom Reports', 'Device Management', 'Driver Management', 'Endpoint Protection', 'Hardware - CD-ROM', 'Hardware - Disk', 'Hardware - General', 'Hardware - Memory', 'Hardware - Modem', 'Hardware - Network Adapter', and 'Hardware - Processor'. A search bar at the top right contains the text 'Location: /ConfigMgr_CAS'.

21. Select the last folder "{<randomnumber>}" and click OK and then Apply.

The screenshot shows a 'Select Data Source' dialog box. It displays a tree view of report categories, similar to the one in the previous screenshot. The last item in the list, '{5C6358F2-4BB6-4a1b-A16E-8D96795D8602}', is highlighted with a blue selection box. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

At this point, you can go back and run the report. You will be prompted for the Collection Name. Select the Collection Name from the drop-down menu and click View Report to display the results.

PART III

Configuration Manager deployment tips

Operating system deployment is a widely-used capability of Microsoft System Center 2012 Configuration Manager. The single chapter in Part III provides tips that cover a wide variety of different scenarios to help you ensure the success of your deployments.

This page left intentionally blank.

CHAPTER 6

Operating system deployment tips

Microsoft System Center 2012 Configuration Manager provides a highly flexible, automated solution to fully deploy and configure servers and desktops from any initial state, including bare-metal deployments. This enables IT administrators to provide an end-to-end solution for the installation and configuration of Windows, by delivering applications, updates, patches, and security fixes in a single distribution. This chapter covers some tips and tricks to operating system deployment.

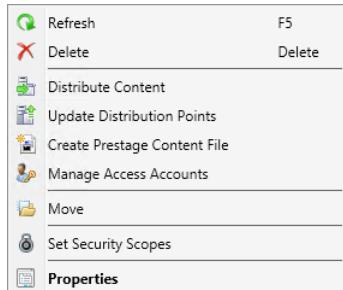
Boot images

The Windows ADK is used to create Windows PE boot images that are used in Configuration Manager to deploy operating systems via PXE, CD-ROM, USB, or hard disk. Windows PE boot images are available in four current versions: 3.0, 3.1, 4.0, and 5.0. Windows PE 4.0 boot images do not work on some older hardware or versions of ESX or VMware Workstation that either do not support Windows 8 or do not have support for NX, PAE, and SSE2. As of Configuration Manager 2012 SP1 CU2, the ability to add Windows PE 3.1 boot images is now available to support these limitations.

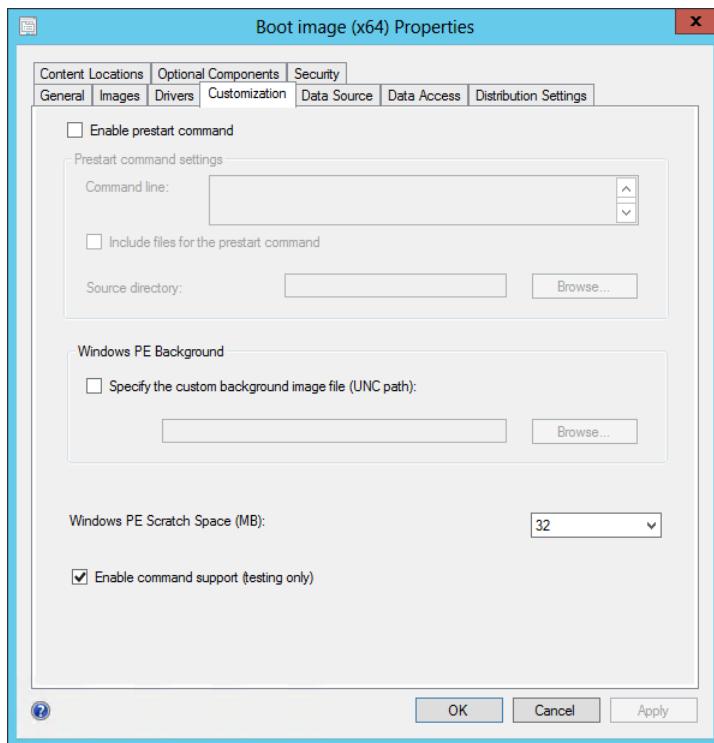
Enabling F8 command prompt support

F8 command prompt support is an essential testing and troubleshooting tool that should be added to all of your boot images. Without it you will not be able to validate IP settings, disk configurations, drivers, or log files to assist in troubleshooting. Below are the steps required to configure F8 command prompt support in your boot images.

1. Open the Configuration Manager console and expand Software Library, Operating Systems, Boot Images, right-click your boot image, and then select Properties.



2. Click the Customization tab, select the Enable Command Support (Testing Only) check box, and then click OK to save and Yes on the next dialog to distribute to the distribution points.



Boot image driver management

Just like operating system images, boot images require drivers for various components such as network, disk, USB 3.0, video, and so on. Most drivers are included in box depending on the Windows PE and operating system versions being used, but in some cases you will need to add additional drivers. This can be a challenging step, and less is more when it comes

to boot image drivers. You can easily end up with a non-functional boot image if you add incompatible drivers.

Various OEM's offer driver packs with a subset of drivers that you might need for your boot image. It is recommended that you identify your requirements and include only the drivers you need. For example, you might not need RAID controller drivers depending on what you plan to deploy, so including them unnecessarily adds extra storage and administrative overhead for testing. It is also best to test against a lab server or non-production boot image before updating drivers to avoid extensive downtime while in production deployments.

Verifying missing drivers in your boot images can sometimes be challenging. One way to make this process easier is to use F8 command line support. Once you have a command prompt you can take advantage of useful functions like diskpart to validate that your Mass Storage Controller is working properly, you can use ipconfig to verify that your network stack is functioning, and you can test missing drivers with drvload to see if you have the correct driver to add to your boot image. Below is a quick step-by-step procedure for using drvload to verify that you have the proper driver.

1. Determine the driver that is missing and download it to media.
2. Insert media and change directory to the folder holding your INF files.
3. Run drvload.exe NameOfYour.inf.
4. Validate that you can access the disk or obtain an IP address
5. Add to a boot image in Configuration Manager.

Optional components

Windows PE Optional Components or Feature Packages allow you to extend Windows PE with additional languages and functionality. By keeping the feature packages external, the footprint remains small and you can tailor each boot image to your own organizational needs. For a complete list of optional components, see the following article: <http://technet.microsoft.com/en-us/library/hh824926.aspx>.

The following optional components might be included by default, but this will vary depending on your needs:

- Windows PowerShell (WinPE-DismCmdlets)
- Storage (WinPE-EnhancedStorage)
- HTML (WinPE-HTA)
- Windows PowerShell (WinPE-StorageWMI)
- Microsoft .NET (WinPE-NetFx4)
- Windows PowerShell (WinPE-PowerShell3)
- Recovery (WinPE-WinReCfg)

Keep in mind that the smaller the footprint of your boot images the faster they will load and the less memory will be required to cache the contents, so don't add everything just in case you *might* use it down the road.

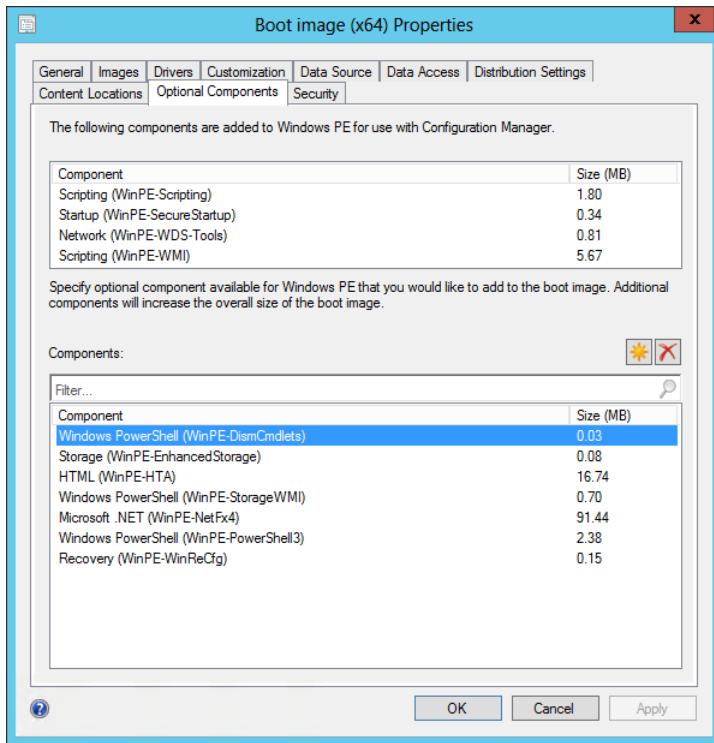


FIGURE 6-1 Windows PE Optional Components.

Adding Windows PE 3.1 to Configuration Manager 2012 SP1 CU2

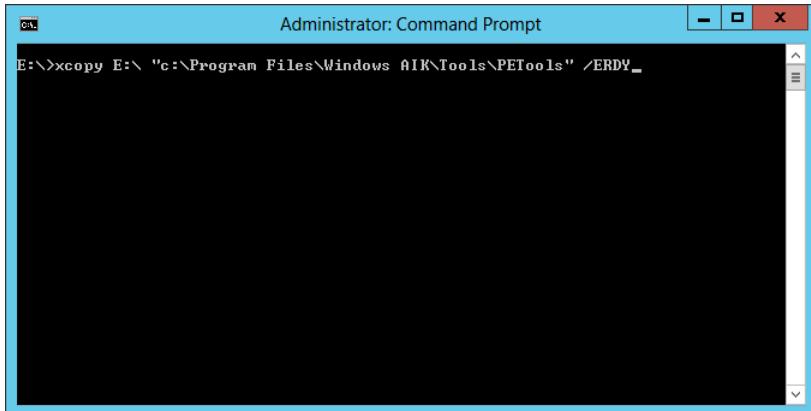
The following procedure shows how to add Windows PE 3.1 to Configuration Manager 2012 SP1 CU2.

1. Install the WAIK 3.0 package on a remote system.

NOTE It is not recommended to have the ADK and WAIK installed on the same system.

2. Install the WAIK 3.1 Supplement update package. To install the supplemental update, open a command prompt and type the following command:

```
E:\>xcopy E:\ "c:\Program Files\Windows AIK\Tools\PETools" /ERDY
```



```
E:\>xcopy E:\ "c:\Program Files\Windows AIK\Tools\PETools" /ERDY-
```

NOTE A total of 1,131 files should be copied.

3. Open the Deployment Tools Command Prompt.
4. Follow the guide on how to create a custom boot image at: [http://technet.microsoft.com/en-us/library/dd744533\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/dd744533(WS.10).aspx)

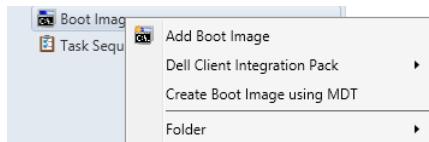
NOTE Be sure to add the following required components:

- WinPE-Scripting
- WinPE-WMI
- WinPE-WDS-Tools
- WinPE-WDS-Tools

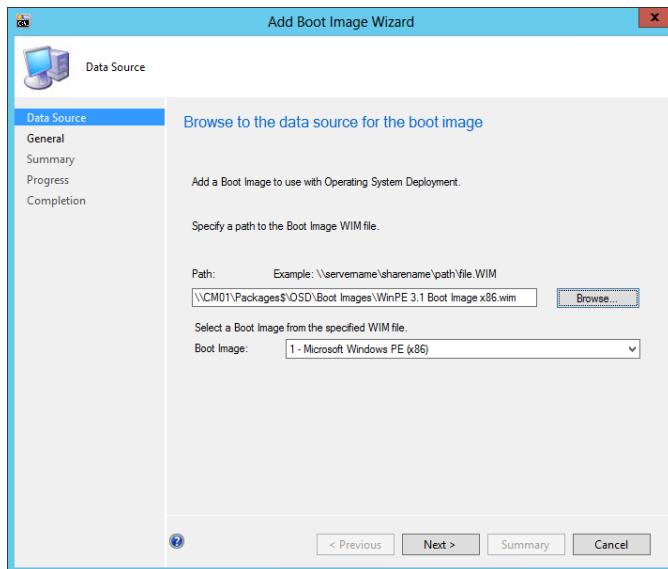
5. Optionally, add drivers to the WIM image before completing step 6.

MORE INFO For more information, see [http://technet.microsoft.com/en-us/library/dd744355\(v=WS.10\).aspx#AddDriverDISM](http://technet.microsoft.com/en-us/library/dd744355(v=WS.10).aspx#AddDriverDISM).

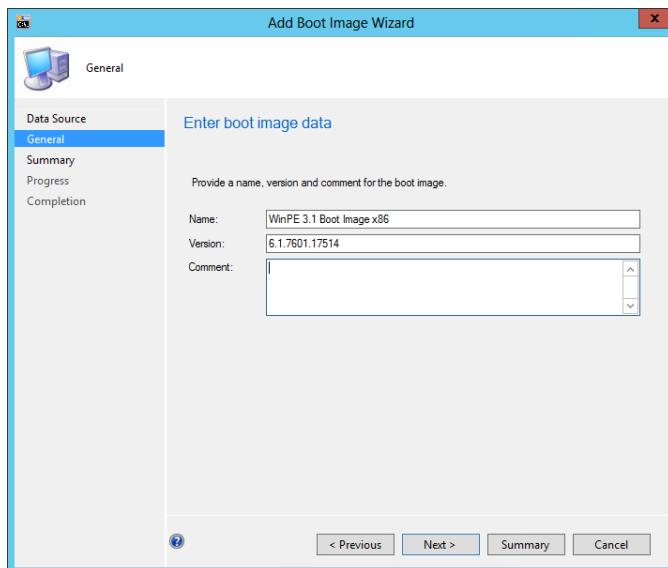
6. Copy the boot.wim file created in step 4 to your Configuration Manager server.
7. Name the boot image appropriately for good housekeeping, *WinPE 3.1 Boot Image x86*.
8. Open the Configuration Manager console and expand Software Library, Operating Systems, Boot Images.
9. Select Add Boot Image.



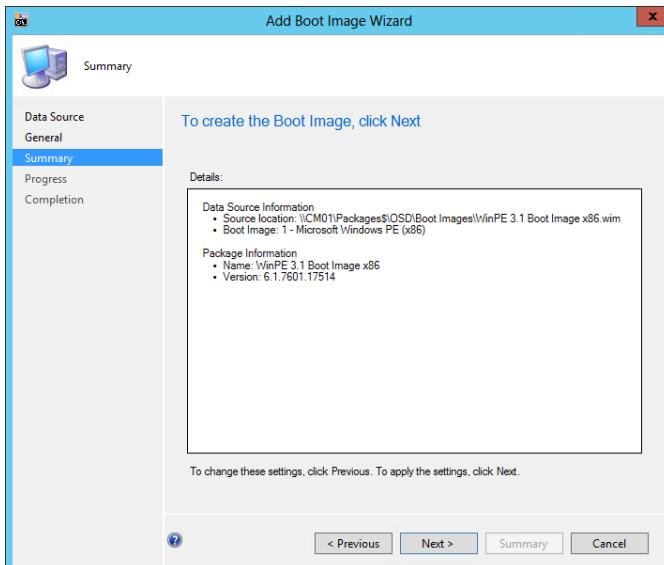
10. Specify the UNC Path to your custom boot image and click Next.



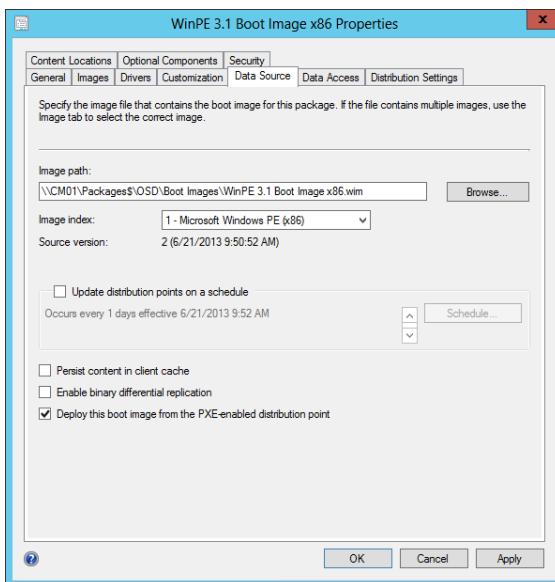
11. Type the Name and Version of your boot image and then click Next.



12. Click Next on the Summary page.



13. Next deploy the boot image to PXE-enabled distribution points by selecting the 'Deploy This Boot Image From The PXE-Enabled Distribution Point' checkbox in the boot image properties dialog box.



You should now have a functional Windows PE 3.1 boot image to use with your legacy systems.

NOTE Some tabs will be unavailable in the boot image properties dialog box; this is normal because the following options are *not supported* with Windows PE 3.1 boot images:

- Installing additional optional components through Configuration Manager
- Adding drivers
- Setting scratch space
- Configuring a prestart command
- Configuring a background image file
- Enabling and disabling command prompt support (debug mode)

Drivers

Driver management in Configuration Manager is one of the most debated topics in operating system deployment and for good cause. It can easily become one of the biggest pain points you face. Best practice is to build a solid driver store that is easy for you to maintain and to know what you have in your environment. Michael Niehaus (<http://blogs.technet.com/b/mniedaus/default.aspx?PageIndex=1>) and Johan Arwidmark (<http://deploymentresearch.com/Research.aspx>) both have very detailed blog posts on the subject with examples of sound driver stores if you would like to find out more information.

Driver signing

For 64-bit versions of Windows Vista and later versions of Windows, driver code signing policy requires that all driver code have a digital signature. In addition, certain configurations of 32-bit versions of Windows Vista and later versions of Windows also require driver code to be digitally-signed in order to access next generation premium content that is controlled by the content protection policy.

You would think that all OEMs are signing their drivers, but sadly this is not the case. You might also encounter and issue with digital signatures when you alter the driver to adjust some of the settings enabled or disabled by default. By doing this you are essentially breaking the digital signature and the driver must be re-signed with a trusted certificate in your environment or by the vendor again. You can sign the drivers yourself, but it is not recommended to the average administrator. However, if you are interested in doing so, you can find great resources online with detailed instructions on the process.

Driver maintenance

Good housekeeping seems like a chore, but untangling a web of disorganization can certainly consume a lot more of your time. You could spend hours finding all of the right drivers for use in your organization. If you spend that much time gathering drivers, you should invest in a structure that is easy to manage and maintain so you can know at a glance what's in your

driver store and be able to add or remove models or drivers on a whim. Figure 6-2 shows a sample folder structure to get started with your driver store. The structure used here is:

Operating System\OEM\Model\Device Version

```
E:\Packages\OSD\Drivers\Windows 8\ASUS
E:\Packages\OSD\Drivers\Windows 8\DELL
E:\Packages\OSD\Drivers\Windows 8\HP
E:\Packages\OSD\Drivers\Windows 8\Lenovo
E:\Packages\OSD\Drivers\Windows 8\Microsoft
E:\Packages\OSD\Drivers\Windows 8\Samsung
E:\Packages\OSD\Drivers\Windows 8\Toshiba
E:\Packages\OSD\Drivers\Windows 8\DELL E6410
E:\Packages\OSD\Drivers\Windows 8\DELL E6510
E:\Packages\OSD\Drivers\Windows 8\HP\HP EliteBook 8570w Mobile Workstation
E:\Packages\OSD\Drivers\Windows 8\Lenovo\Lenovo ThinkPad W530
E:\Packages\OSD\Drivers\Windows 8\Microsoft Surface Pro
```

FIGURE 6-2 Sample folder structure for drivers.

It's important to regularly check your OEM for updated drivers not only to improve device functionality but to also patch security vulnerabilities in the drivers.

UEFI

The Unified Extensible Firmware Interface (UEFI) is meant to replace the legacy BIOS interface of today. UEFI Spec 2.3.1 is the latest version available and is a requirement for Windows 8 certified machines. There are two classes of UEFI: class 2 devices, meaning they support the BIOS compatibility mode known as Compatibility Support Module (CSM), and class 3 devices, which are native UEFI systems only. The advantages of UEFI include:

- Full memory access (32-bit or 64-bit)
- CPU independence (enabling UEFI on x86, x64, ia64, and even ARM computers)
- Faster initialization
- Support for larger boot disks (larger than 2.2 terabytes)

The following are a few tips concerning UEFI support in Windows:

- Native class 3 UEFI devices support only Windows 8 and newer.
- Boot images must be the same architecture as the operating system being deployed.
- Use the _SMSTSBootUEFI task sequence variable to determine if a machine is in UEFI mode.
- Always build your reference image on a virtual machine in BIOS compatibility mode.
- Server 2012 or higher is required for UEFI x86.
- Server 2008 R2 or higher is required for UEFI x64.
- USB boot media must be formatted as FAT32.
- FAT32 has a 4-GB file size limit.

The typical disk layout for UEFI deployments looks like this:

- Recovery (NTFS, 300 MB)
- System (FAT32, active, 300 MB)

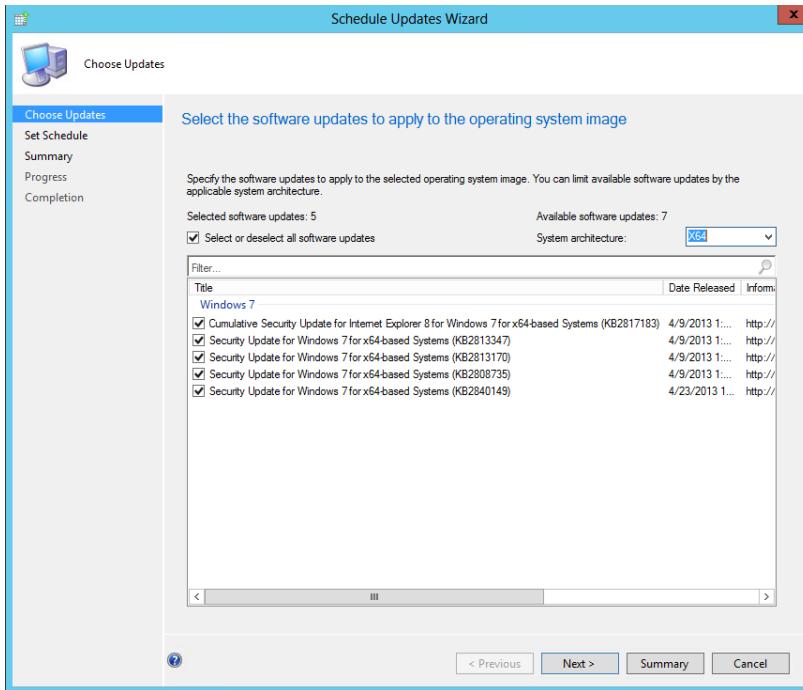
- Microsoft Reserved Partition (MSR, 128 MB)
- Operating System (NTFS, 100%)

Operating System Images and Offline Servicing

Windows image files, or WIMs, can be imported into Configuration Manager as Operating System Images. This can be in the form of the original install.wim from the product source ISO or DVD, or it can be your own custom WIMfile. WIMs have a catalog file associated with them that shows the state of all settings and packages in the image file. Additionally, an unattend-ed.xml file can be used to control the state of the components, packages, and settings.

Configuration Manager 2012 includes a feature for applying updates to Operating System Images called Offline Servicing. This feature in the Operating System node under Software Library allows you to apply Component Based Servicing (CBS) updates to a WIM file offline. This allows for faster deployment times, reduced image recycle periods, and overall lower administrative effort to maintain Operating System Images. Following is an example of how to use this feature.

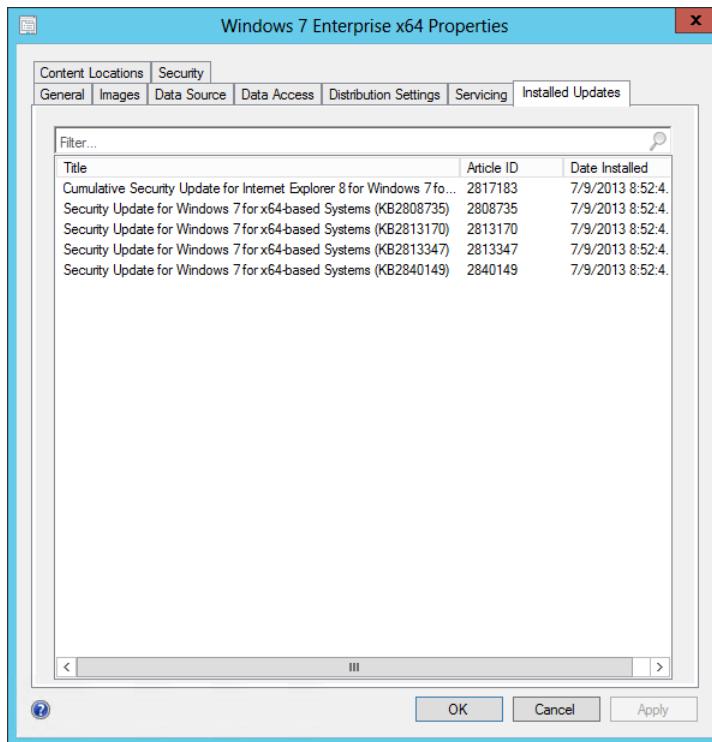
1. Open the Configuration Manager console and expand Software Library, Operating Systems, Operating System Images, and then select Schedule Updates from the ribbon.
2. Review the list of updates that meet the following criteria:
 - They are CBS updates.
 - They have been deployed in your Configuration Manager environment. (This helps limit the selection of updates to those that have been deployed or tested in your enterprise.)
 - They have not successfully been deployed to your image file through the Offline Servicing feature.
3. The updates are filtered to match the architecture of your image file. Select your updates and then click Next.



4. Unless you want to schedule the updates for a later time, accept the defaults and click Next, click Next again, and then click Close.
5. You can validate the progress by watching OfflineServicingMgr.log on the site server.

```
Completed processing image package P0100115. Status = Success
STATMSG: ID=7904 SEV=I LEV=M SOURCE="SMS Server" COMP="SMS_OFFLINE_SERVICING_MANAGER" SYS=CM01.CC
Create backup copy for image P0100115
copying image file "\CM01\Packages$\OSD\Operating Systems\Windows 7 Enterprise x64\install.wim" to "\CM01\Packa
Copy image (ID P0100115, size 2632 MB)
copying file from 'D:\ConfigMgr_OfflineImageServicing\P0100115\install.wim'
    to "\CM01\Packages$\OSD\Operating Systems\Windows 7 Enterprise x64\install.wim" ....
Copying (25% complete)...
Copying (50% complete)...
Copying (75% complete)...
Copying (100% complete)...
Original image '\CM01\Packages$\OSD\Operating Systems\Windows 7 Enterprise x64\install.wim' is backed up at '\CM
Successfully requested image P0100115 to be updated from its source.
```

6. To see all of the updates that have been added to your image, review the Installed Updates tab in the image file's properties dialog box.



Task sequences

Task sequences provide the mechanism for performing multiple steps or tasks on a client computer at the command-line level without requiring user intervention. Task sequences do not represent a full scripting language but are highly customizable and include a large subset of built-in actions and pre-configured templates.

MDT integration

Microsoft Deployment Toolkit (MDT) is a Solution Accelerator for operating system and application deployment. MDT supports deployment of client and server operating systems. It can be used as a standalone or integrated into Configuration Manager. By integrating MDT into Configuration Manager, you gain the ability to use the additional templates, actions, and customized boot images and to call external scripts, databases, web services, and much more to enhance and streamline your operating system deployment needs.

To integrate MDT with Configuration Manager, install MDT on your Configuration Manager site server or on any computer with the console installed that you might use to edit a task sequence. After you install MDT, follow the simple Configure ConfigMgr Integration Wizard, shown in Figure 6-3.

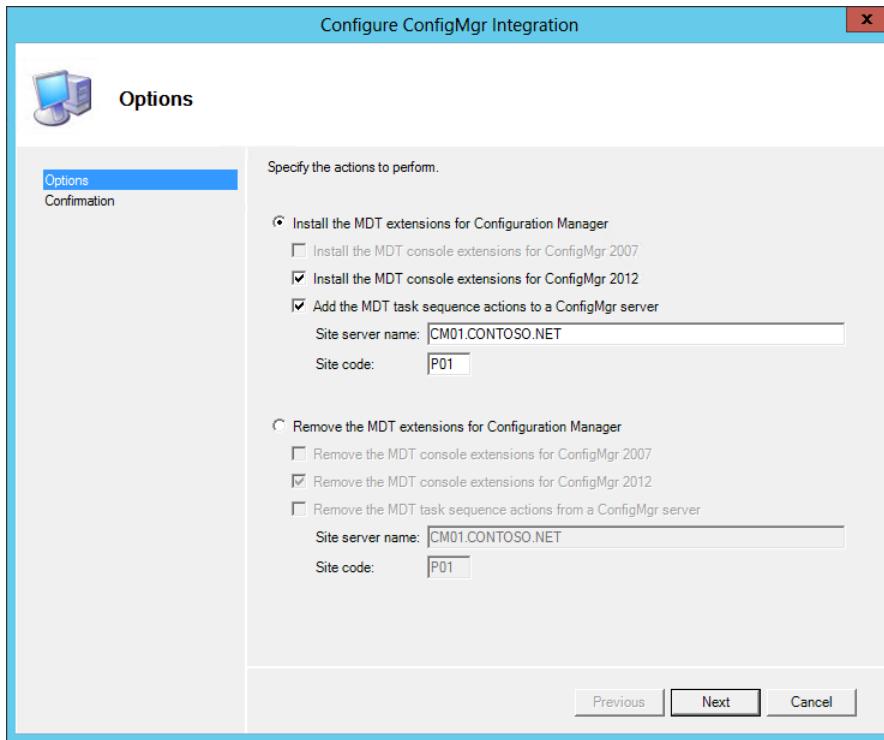


FIGURE 6-3 Integrating MDT with Configuration Manager

MDT integration is relatively simple, requiring a set of binary files and XML extensions to be copied and a Microsoft Operations Framework (MOF) file to be compiled, all of which are done via the wizard during integration. A handful of issues could prevent this from working smoothly, commonly permissions issues, antivirus blocking, or corrupt WMI. After integration, you can open an existing or create a new task sequence to see all of the new options under the MDT node in the task sequence.

Static IP address assignment

Occasionally, you might require the use of static IP addresses. You might simply want to assign static IP addresses to your server systems during deployment, or DHCP is unfortunately nowhere to be found and you must work with what you have to get the job done. Luckily, this requirement isn't too difficult to achieve. You can assign IP addresses on a collection level or via a custom startup script to assign the proper variables. Figure 6-4 shows an example of setting collection variables for a static IP address.

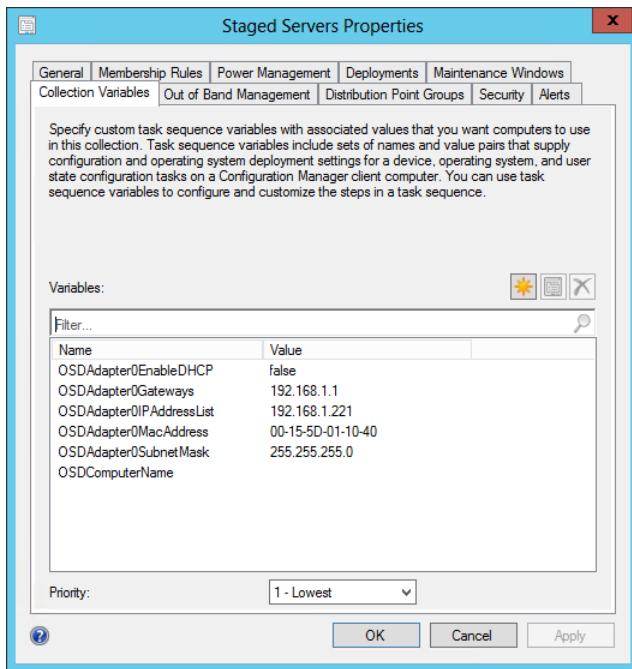
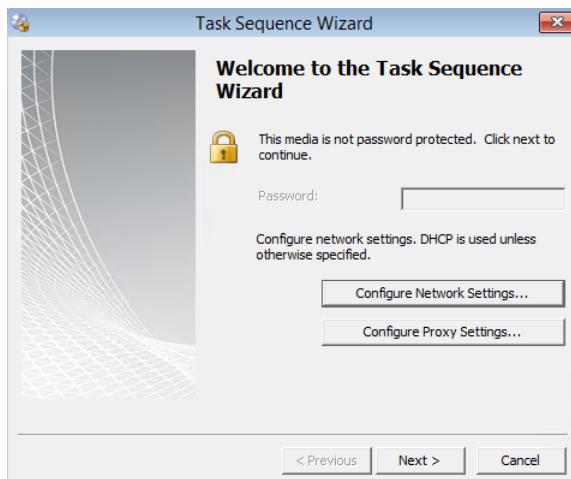


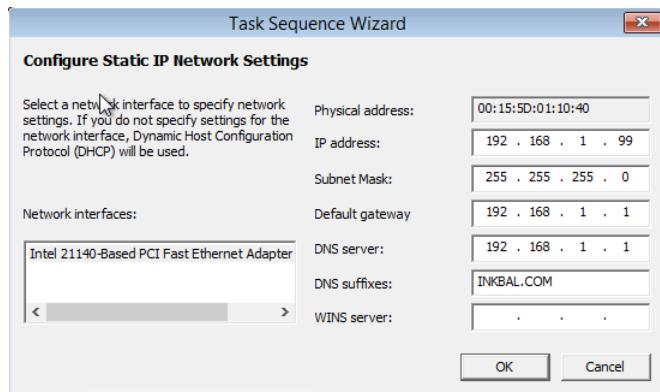
FIGURE 6-4 Configuring collection variables for a static IP address.

You can also create task sequence bootable media, which will generate a dialog box to enter the above information. Having bootable media is a requirement since it's expected that if you can PXE boot, you already have an IP address and don't need to statically assign an address in this manner.

1. Insert bootable media and click Configure Network Settings on the first page of the Task Sequence Wizard.



- Supply your IP settings and click OK.



- Verify that the IP address was assigned correctly by viewing the X:\Windows\Temp\SMSTSLog\SMSTS.log file.

```

Enumerating available network adapters.
Reading adapter information for "Intel 21140-Based PCI Fast Ethernet Adapter (Emulated)"
Applying settings for Intel 21140-Based PCI Fast Ethernet Adapter (Emulated)
SetGateway("192.168.1.1")
EnableStatic("192.168.1.99", "255.255.255.0")
SetDNSServerSearchOrder("192.168.1.10")
SetDNSuffixSearchOrder("INKBAL.COM")
SetGateway("192.168.1.1")
Executing command line: X:\windows\system32\cmd.exe /k
  
```

OSDPreserveDriveLetter

Configuration Manager 2012 SP1 includes a new variable to control the behavior of the drive letter assignment. This variable determines whether or not the task sequence uses the drive letter captured in the operating system image WIM file when applying that image to a destination computer.

Setting OSDPreserveDriveLetter=YES will deploy the operating system to the drive from which it was captured (see Figure 6-5). For instance, the install.wim is captured with the c:\ drive as the destination disk, so that would hold true with OSDPreserveDriveLetter set to true. In some cases, this setting does not produce the desired affect, but in most cases this will work fine.

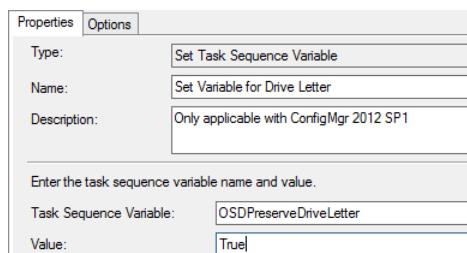


FIGURE 6-5 Setting the variable for the drive letter.

SMSTSPostAction

The SMSTSPostAction task sequence variable is a very handy addition and gives you the ability to specify a command to be run after the task sequence completes. For example, you can use this variable to specify a reboot action after the task sequence deploys an operating system to the device (see Figure 6-6).

Typically you would just add a simple restart computer step to a task sequence, but if you are trying to get Group Policy to process faster, this would not work since the task sequence disables Group Policy Object (GPO) processing during the task sequence. The SMSTSPostAction gets around this implication and offers many other possibilities as you can likely imagine. Additionally, it does not matter where you set this variable in the task sequence since it will be processed at the end regardless of location.

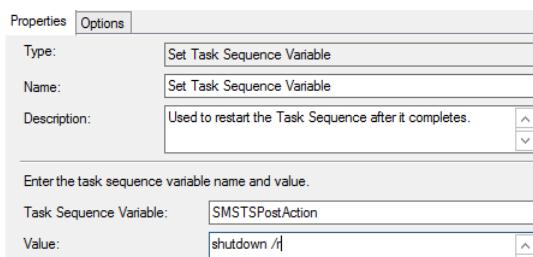


FIGURE 6-6 Configuring the task sequence to restart after it completes.

Extending task sequence logging

Logging in Configuration Manager can certainly be overwhelming at times. There are log files for just about everything, but the primary log file for operating system deployment is the SMSTS.log file, which will grow and grow and grow, depending on the size of your task sequence. One challenge is finding what you are looking for in the log, which can be complicated by the fact that once the log fills up, it rolls over and a new log file is created. By default only one rollover log is maintained and the rest of the log is purged.

When it comes to logging, sometimes more is better. Luckily, the Configuration Manager team has provided the ability to extend logging so you can have more to work with if you are troubleshooting something or simply want to verify what happened. CCMLOGLEVEL, CCMLOGMAXHISTORY, and CCMLOGMAXSIZE all are configurable in the installation properties of the ConfigMgr Client Installation step of the task sequence. By setting these variables, you will gain more robust logging (see Figure 6-7):

- CCMLOGLEVEL=0 Turns on verbose logging.
- CCMLOGMAXHISTORY=6 Sets the number of log files to keep.
- CCMLOGMAXSIZE=2621440 Sets the log size to 2.5 MB.

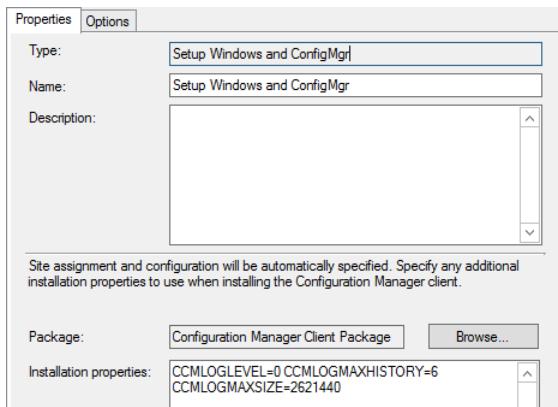


FIGURE 6-7 Configuring logging options.

Applications

Applications in Configuration Manager give you as the administrator flexibility in application delivery. An application can be targeted either at a user, at a device, or at the user's device without having to know what the device name is before deployment. The application model has very granular control with requirements, dependencies, supersedence, revision control, and conditions.

Chassis type global conditions

Global conditions in Configuration Manager 2012 are rules that represent business or technical conditions that can be used to specify how an application is provided and deployed to client devices. Global conditions can be created from the Global Conditions node of the Configuration Manager console or from within the Create Deployment Type Wizard. Global conditions are accessed and used from the Requirements page of the Create Deployment Type Wizard. With the ability to target applications to laptops, desktops, or servers, you can leverage the flexibility of Global Conditions within the new application model of System Center 2012 Configuration Manager.

When leveraging requirements within the application model, you can select a global condition for chassis type, which will then be evaluated before the content is downloaded from the server. This feature precludes the requirement for a collection evaluation or refresh unlike setting these types of criteria on collection queries. As a result, the process is more efficient than in prior versions of Configuration Manager.

MORE INFO For details about how to set up chassis-type global conditions, go to <http://blogs.technet.com/b/brandonlinton/archive/2013/01/30/configmgr-2012-chassis-type-global-condition.aspx>.

Installing applications that require interaction

Sometimes you run into a stubborn application that just won't install during your task sequence. Perhaps you have gone down the road of initial troubleshooting by trying to install the application outside of the task sequence as a package or application and found that works fine. Perhaps you even tried manually running and the application works without a hitch. So why can't you get this stubborn application to install during your task sequence?

In Windows Vista and above a security feature was implanted to isolate services in session 0 and make them non-interactive. Unfortunately, Configuration Manager task sequences run under the system context in session 0, so any application that requires interaction will fail or simply sit and time out. Luckily, the MDT Solution Accelerator team thought of this for User Defined Installations (UDI) and developed a handy executable that launches in a user session and makes the process interactive and visible to the technician and task sequence.

The following example shows how to launch an executable with ServiceUI.exe from the Microsoft Deployment Toolkit:

```
ServiceUI.exe -process:tsprogressui.exe YourApplication.exe
```

Application logging

One thing that is frequently overlooked is logging when packages and applications are created and put into Configuration Manager. Because this isn't a requirement and because most of us are overworked, this step is easily forgotten event though it is essential to good package procedures and troubleshooting efforts.

Most common installers, such as InstallShield and Wise, offer logging capabilities in their applications, which can be used for advanced troubleshooting when things don't go right. Here are a couple of quick examples:

```
Msiexec /I yourapplication.msi /qb! /l*v "%WINDIR%\Temp\YourApplication.log"  
Setup.exe /s /log "%WINDIR%\Temp\YourApplication.log"
```

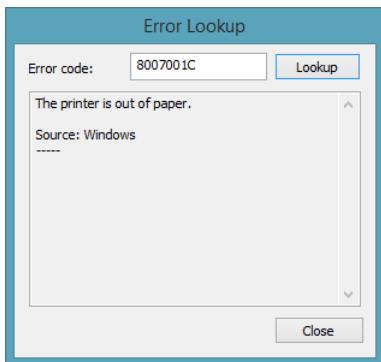
Troubleshooting

Operating system deployment troubleshooting is a "Where's Waldo" of log files. With so many different logs and locations to look for things, it becomes a talent to master just the troubleshooting end of operating system deployment. Even after finding the logs and their locations, how do you decipher the detailed output in the logs? Luckily, an extensive list of online TechNet documents list all of the log files and what they are for. Now you just need to know how to use the best tools to read them, have a basic understanding of what to look for and where, and, during testing, set up central logging to make things easier on failed deployments.

CMTrace

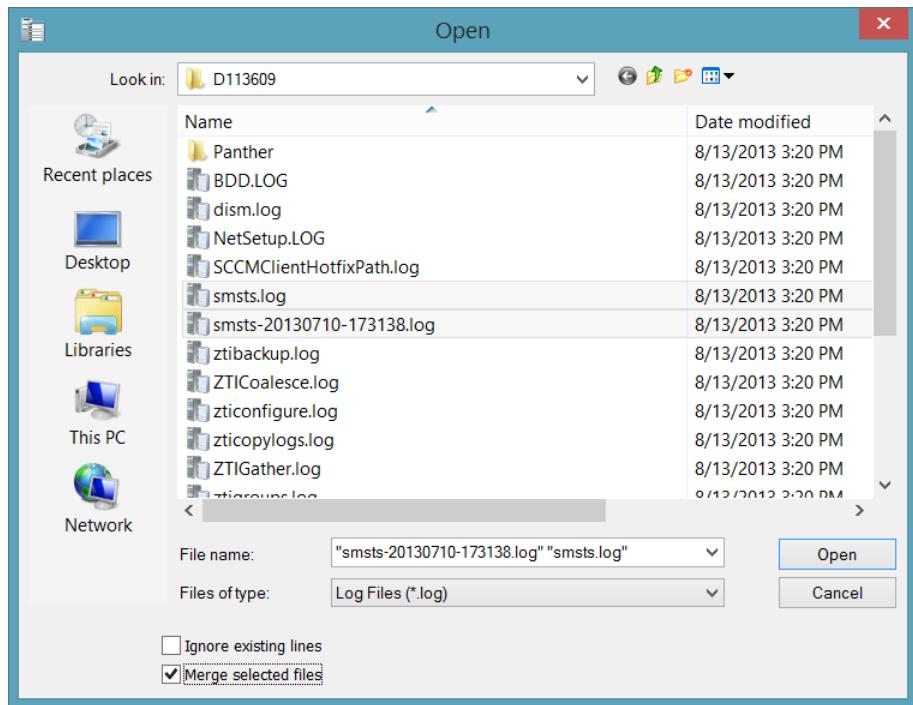
The Configuration Manager Toolkit includes a utility called CMTrace.exe that is an advanced log file viewer that can open and read live log files and can also merge multiple log files into a single file with sorting options and error code lookups. Additionally, CMTrace will highlight certain parts of the log, such as errors, failures, warnings, and informational messages with the appropriate colors to make it easier to find information in the log file. Follow these steps to use CMTrace.

1. Open CMTrace.exe and find the error code that you want to research.
2. Select the Error Lookup option from the Tools menu or press CTRL+L.
3. Type the error and you should see a friendly, formatted message:

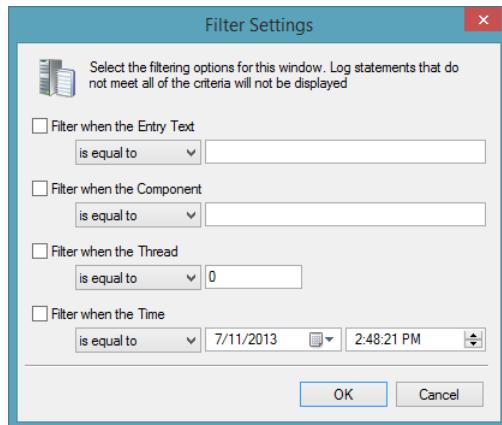


Merging log files can be incredibly beneficial when trying to trace the flow of an issue. Follow these steps to use CMTrace to merge log files.

1. Open CMTrace.exe and then press CTRL+O to access the Open dialog box.
2. Select the files you want to merge, select the Merge Selected Files check box, and click Open.



3. Now that you have a merged log file, you can also apply a filter to find just what you are looking for in the merged log files. Select Tools, Filter to access the filter settings (shown in the following screenshot).



Log locations

Configuration Manager log files are stored in a variety of locations, depending on the process that creates the log file. Table 6-1 lists some typical log locations where you might find log files depending on what point the system is at in the deployment process.

TABLE 6-1 Typical log file locations

Scenario	Typical location
WinPE before formatting HDD	X:\Windows\Temp\SMSTSLog
WinPE after HDD is formatted	X:\SMSTSLog C:_SMSTaskSequence\logs\smstslog
Windows OS during Task Sequence execution	C:_SMSTaskSequence\Logs\SMSTSLog
Windows OS after ConfigMgr Agent is installed and initialized	%WINDIR%\system32\CCM\Logs\SMSTSLog
After Task Sequence Execution is completed	%WINDIR%\CCM\Logs
Workgroup and Domain join log	%WINDIR%\debug\NetSetup.LOG
Windows Setupact and Setuperr logs	%WINDIR%\Panther

Centralized logging

As mentioned earlier, MDT Integration gives you a lot of flexibility and control during operating system deployment. When it comes to troubleshooting log files you can see that there are a number of logs and a number of locations to find these logs, which can be daunting when under pressure.

MDT has a variable called SLShare that you can use in MDT Integrated task sequences to find all of the common log files and copy them up to a file share when the task sequence is complete. This makes troubleshooting a breeze as you can simply open your file share and review the logs that have been copied and placed in a folder named for the machine that was deployed.

Setup is simple: just create a share on your site server or somewhere on your network and set the SLShare variable in your task sequence or customsettings.ini file. Once completed, folders will be created under the Logs directory for each machine, like the example shown in Figure 6-8, which uses the following share:

```
SLShare=\\CM01\\Packages$\\OSD\\Logs
```

Name	Date modified	Type	Size
Panther	7/11/2013 2:22 PM	File folder	
BDD.LOG	7/11/2013 2:22 PM	LOG File	107 KB
dism.log	7/11/2013 2:22 PM	LOG File	537 KB
NetSetup.LOG	7/11/2013 2:22 PM	LOG File	13 KB
SCCMClientHotfixPath.log	7/11/2013 2:22 PM	LOG File	3 KB
smsts.log	7/11/2013 2:22 PM	LOG File	457 KB
smsts-20130710-173138.log	7/11/2013 2:22 PM	LOG File	977 KB
ztibackup.log	7/11/2013 2:22 PM	LOG File	1 KB
ZTICoalesce.log	7/11/2013 2:22 PM	LOG File	2 KB
zticonfigure.log	7/11/2013 2:22 PM	LOG File	19 KB
zticopylogs.log	7/11/2013 2:22 PM	LOG File	4 KB
ZTIGather.log	7/11/2013 2:22 PM	LOG File	46 KB
ztigroups.log	7/11/2013 2:22 PM	LOG File	3 KB
ztimovestatestore.log	7/11/2013 2:22 PM	LOG File	1 KB
ZTINicConfig.log	7/11/2013 2:22 PM	LOG File	3 KB
ztioptin.log	7/11/2013 2:22 PM	LOG File	1 KB
ZTISCCM.log	7/11/2013 2:22 PM	LOG File	18 KB
ztisetbackground.log	7/11/2013 2:22 PM	LOG File	4 KB
ztitatoo.log	7/11/2013 2:22 PM	LOG File	2 KB
ZTIValidate.log	7/11/2013 2:22 PM	LOG File	2 KB

FIGURE 6-8 OSD log files.

About the authors

Rushi Faldu



Rushi Faldu, a Senior Premier Field Engineer supporting Microsoft System Center 2012 Configuration Manager, has been with Microsoft for eight years. He has been working with the product since SMS 2.0. He is a lead for System Center Concepts & Administration workshop, which is delivered to many of premier customers throughout the world. He is also an author of two ebooks, *Microsoft System Center: Troubleshooting Configuration Manager* and *Microsoft System Center: Configuration Manager Field Experience*. Rushi resides in New Jersey and enjoys P90X and Insanity workouts in his free time. He loves hiking, camping, and playing tennis with his daughters.

Manish Raval



Manish Raval is a Consultant with Microsoft Consulting Services (MCS), based in Calgary, Canada. Manish has more than eight years of IT experience. He started his career at Microsoft in the support group supporting Active Directory and later SMS 2003 and Microsoft System Center Configuration Manager2007, both as a Support Engineer and a Trainer and Technical Lead.

Manish is passionate about working with customers. He specializes in Windows Server Active Directory, Virtualization, Desktop Deployment, and the System Center suite of products with a special emphasis on System Center Configuration Manager. He has also spoken at TechEd India. In his free time, he enjoys seasonal sports like skiing, biking, hiking, swimming, and so on.

Brandon Linton



Brandon Linton is a Senior Consultant with Microsoft Consulting Services, with over 12 years of IT experience. He specializes in Microsoft System Center products and operating system deployment. Brandon is actively involved in communities such as myitforum.com and TechNet Forums, and is also an active blogger. Brandon resides in North Eastern Pennsylvania with his wife Lori and son Mason. When not working within the technical communities or on solutions, he can be found in audio/video forums and working on his home theater.

Kaushal Pandey



Kaushal Pandey works as a Consultant with Microsoft Consulting Services. Kaushal has more than 8 years of IT experience. He started his career working with SMS 2003 and has seen the technology evolve from Systems Management Server to Microsoft System Center Configuration Manager. He is passionate about technology, and in his free time he loves to play cricket or swim.

Manoj Kumar Pal (contributing author)

Manoj Kumar Pal is a Consultant with Microsoft Consulting Services with more than 10 years of IT experience. His area of expertise are the Microsoft System Center suite and client management. He is passionate about technology, and in his free time he loves to play games on Xbox. Manoj earned his computing degree in India. He is lives in Chicago with his wife and son.

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 or contributed to over two dozen books, including *Windows 7 Resource Kit* (Microsoft Press, 2009), for which he was lead author; *Understanding Microsoft Virtualization Solutions: From the Desktop to the Datacenter* (Microsoft Press, 2010); and *Introducing Windows Server 2012* (Microsoft Press, 2012), a free ebook that has been downloaded almost three quarters of a million times.

Mitch has been repeatedly awarded Most Valuable Professional (MVP) status by Microsoft for his outstanding contributions to supporting the global IT community. He is a nine-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 (<http://www.wservernews.com>), a weekly newsletter focused on system administration and security issues for the Windows Server platform. With more than 100,000 IT pro subscribers worldwide, WServerNews is the largest Windows Server–focused newsletter in the world.

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>.

You can also follow Mitch on Twitter at <http://twitter.com/mitchtulloch> or like him on Facebook at <http://www.facebook.com/mitchtulloch>.



Now that
you've
read the
book...

Tell us what you think!

Was it useful?

Did it teach you what you wanted to learn?

Was there room for improvement?

Let us know at <http://aka.ms/tellpress>

Your feedback goes directly to the staff at Microsoft Press,
and we read every one of your responses. Thanks in advance!



Microsoft