Operations Guide

Microsoft® Systems Management Server 2003

Scalable Management for Windows-based Systems

Microsoft

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 1994-2003 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, Windows, Windows NT, Active Directory, Intellimirror, Microsoft Press, Win32, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Document No. X09-75018
Printed in the United States of America.

Contents

Getting Started . . . . . **xix**
Technical Resources xix
Online Library xix
Product Documentation Available for SMS xx
Keeping Your Technical Knowledge Current xxi
Document Conventions xxi
PART 1 . . . . . **1**
CHAPTER 1 Scenarios and Procedures for Deploying SMS 2003 . . . . . **3**
Overview of the Deployment Process 4
Client Support 5
SMS Deployment Components 7
Part 1: Hierarchy-Specific Questions 9
Upgrade Questions 10
Options for Client Migration 13
Active Directory Questions 15
Network Questions 17
Part 2: Site-Specific Questions 19
Site Configuration Questions 19
Client Configuration Questions 23
Part 3: SMS 2003 Deployment Scenarios 27
New Installation 27
Central Site Installation 28
Client Installation 29
Management Point Installation 32

In-Place Upgrade . . . 33
In-Place Upgrade Deployment Steps . . . 33
Upgrade Site . . . 35
Side-by-Side Upgrade . . . 38
Post-Installation Considerations . . . 40
CHAPTER 2 Collecting Hardware and Software Inventory . . . 43
Hardware Inventory Administrative Tasks . . . 45
Enabling and Disabling Hardware Inventory . . . 45
Scheduling Hardware Inventory . . . 46
Enabling and Disabling MIF Collection . . . 47
Configuring Hardware Inventory Rules . . . 48
Editing SMS_def.mof . . . 49
Distributing SMS_def.mof . . . 51
Upgrading SMS and SMS_def.mof . . . 51
Software Inventory Administrative Tasks . . . 52
Enabling and Disabling Software Inventory . . . 53
Scheduling Software Inventory . . . 54
Configuring Software Inventory Rules . . . 54
Configuring File Collection . . . 56
Managing Inventory Names . . . 57
Controlling Software Inventory on Servers . . . 58
Using Resource Explorer to View Inventory Data . . . 59
Viewing Hardware Inventory . . . 59
Viewing Hardware Inventory History . . . 60
Viewing Software Inventory . . . 61
Viewing Collected Files . . . 61
Reviewing the Inventory Data . . . 62
Other Considerations for Collecting Inventory . . . 65
Hardware and Software Inventory Behavior When Clients Cannot Connect to the SMS Site . . . 66
Collection of User Context Information . . . 66
CHAPTER 3 Advanced Inventory Collection . . . 67
Using Resource Explorer from the Command Line . . . 68
Extending Hardware Inventory . . . 69
Creating Hardware Inventory Extensions . . . 70
Propagating Hardware Inventory Extensions Throughout the SMS Hierarchy . . . 70

MIF Extensions . . . 71
Customizing with NOIDMIF Files . . . 71
Creating a Class by Using a NOIDMIF File . . . 72
Customizing with IDMIF Files . . . 75
Requirements of IDMIF Files . . . 75
MOF Extensions . . . 76
Understanding the Relationship Between the Hardware Inventory Agent and WMI . . . 77
Customizing with MOF Files . . . 79
Scripted Extensions . . . 84
Changing or Removing Hardware Inventory Extensions . . . 86
Common MOF Extensions . . . 87
Finding Computers That Are Laptops . . . 87
Finding Computer Serial Numbers . . . 89
Finding Hotfix Information . . . 89
Collecting Windows Installer Information . . . 91
Collecting SQL Server Information . . . 92
CHAPTER 4 Managing Collections and Queries . . . 95
Working with Collections . . . 96
Understanding Collections . . . 96
Collections that Provide Management Scope . . . 98
Subcollections . . . 98
Collections in the SMS Hierarchy . . . 99
Collection and Resource Security . . . 100
Creating and Managing Collections . . . 101
Managing Resources in Collections . . . 104
Working with Queries . . . 107
Understanding SMS Database Classes . . . 107
Understanding SMS Queries . . . 109
SMS Object Types . . . 109
Required SMS Query Elements . . . 111
Optional SMS Query Elements . . . 111
WMI Query Language . . . 115
Creating and Managing SMS Queries . . . 115
Creating and Editing Query Statements . . . 119

CHAPTER 5 Distributing Software **125**
Preparing to Distribute Packages 126
Configuring the Software Distribution Agent 126
Preparing CAPs, Management Points, and Distribution Points 128
Preparing Collections 131
Preparing Security 133
SMS Administrator Console Security 133
Package Access Accounts 133
Legacy Client Software Installation Account 135
Advanced Client Network Access Account 136
Configuring the Software Distribution Component 137
Managing Packages 139
Creating and Managing Packages 139
Create Package Source Directories 140
Create a New Package 141
Create a Setup Script 145
Modify an Existing Package 145
Delete a Package 146
Creating and Managing Programs 146
Create a New Program 147
Modify an Existing Program 154
Delete a Program 155
Distributing Packages 155
Managing Advertisements 159
Creating Advertisements 159
Creating Advertisements with Assigned Programs 161
Assigned Program Scenarios 163
Advertisements to Advanced Clients 164
Disabling or Rerunning Advertisements 165
Ensuring Package and Advertisement Integrity 165
Maintaining Packages and Advertisements 167
Monitoring Software Distributions 168
Using Status Summaries for Packages at Their Sites and Distribution Points 169
Monitoring Package Distribution 169
Monitoring Advertised Programs 170
Using Status MIFs 171

Using Software Distribution Tools and Wizards . . . 172
Running Advertised Programs on SMS Clients . . . 174
Running Advertised Programs on Either Client . . . 175
Running Advertised Programs on Advanced Clients . . . 176
Running Advertised Programs on Legacy Clients . . . 180
Software Distribution Common Practices . . . 182
Software Distribution Best Practices . . . 186
CHAPTER 6 Managing Software Updates . . . 189
Software Update Management Overview . . . 190
About Software Updates . . . 190
About Service Packs . . . 191
Challenges in Managing Software Updates . . . 191
Software Update Management Guidelines . . . 192
How Software Update Management Works . . . 193
Basic Components Functionality . . . 194
Underlying Technology . . . 195
Software Update Management Advanced Features . . . 196
Software Update Management Tasks . . . 197
Preparing for Software Update Management Tasks . . . 198
Task 1: Review the System Requirements for the Software Update Management Components . . . 198
Task 2: Prepare the Test Environment . . . 203
Task 3: Prepare the Production Environment . . . 205
Task 4: Deploy the Software Update Inventory Tools . . . 206
Tasks for Authorizing and Distributing Software Updates . . . 220
Task 1: Prepare the Package Source Folder . . . 221
Task 2: Plan the Software Update Packages . . . 221
Task 3: Evaluate and Prioritize the Software Updates . . . 224
Task 4: Isolate and Test the Software Updates . . . 225
Task 5: Create the Software Updates Packages . . . 225
Notes on Deploying Microsoft Office Updates . . . 231
Task 6: Customize the Package and Advertisement Settings . . . 240
Task 7: Test the Software Update Packages . . . 241
Task 8: Expedite Delivery of New or Urgent Updates (optional) . . . 243

Monitoring Software Update Distributions . . . 244
Tools for Monitoring Software Update Distributions . . . 245
Software Update Reporting . . . 246
Software Update Status Messages . . . 247
Software Update Logging . . . 248
Tasks for Monitoring Software Update Processes . . . 249
Task 1: Audit the Enterprise for Current Security Vulnerabilities . . . 249
Task 2: Monitor the Status of Software Update Distributions . . . 250
Task 3: Check the Health of Software Update Management Components . . . 252
Task 4: Troubleshoot Software Update Installation Errors . . . 253
Software Update Management Best Practices . . . 254
General Best Practices . . . 254
Setup: Best Practices . . . 255
Inventory Synchronization: Best Practices . . . 256
Software Update Inventory: Best Practices . . . 257
Software Update Distribution: Best Practices . . . 258
Software Update Installation: Best Practices . . . 260
End-User Experience: Best Practices . . . 261
Monitoring: Best Practices . . . 262
Scheduling: Best Practices . . . 262
About Scheduling Software Update Installation Advertisements . . . 265
About Updating Distribution Points . . . 265
Performance Considerations . . . 266
Processing Load Added to SMS Client Computers by the Software Update Management Components . . . 266
Inventory Data Considerations . . . 266
Scan Component Bandwidth Considerations . . . 267
Scan Component Completeness Considerations . . . 268
Status Message Processing Considerations . . . 269
Instantaneous Loading Considerations . . . 269
General Cumulative Effect of Scan Tools . . . 269
Resolving Network Issues for Mobile Clients . . . 269
CHAPTER 7 Creating Software Installation Packages with SMS Installer . . . 271
SMS Installer Overview . . . 272
SMS Installer Process . . . 272
SMS Installer Tasks . . . 274

Installing and Starting SMS Installer . . . 275
Repackage Installation Wizard . . . 287
Reference Computer Preparation . . . 288
Running Repackage Installation Wizard . . . 289
Configuring Repackage Installation Wizard . . . 290
Custom Configuration for Repackaging Scans . . . 291
Watch Application Wizard . . . 292
Customizing Scripts with the Script Editor . . . 293
Script Editor User Interface . . . 294
Installation Script Variables . . . 295
Using an Installation Script to Wrap an Existing Setup . . . 303
Testing SMS Installer-generated Executable Files . . . 303
Distributing SMS Installer-generated Executable Files . . . 305
SMS Installer-generated Executable File Compilation . . . 305
PART 2 . . . 307
CHAPTER 8 Software Metering . . . 309
Overview . . . 310
How Software Metering Works . . . 310
Changes to Software Metering . . . 311
Configuring and Using Software Metering . . . 312
Enabling Software Metering . . . 312
Excluding Advanced Clients from Software Metering . . . 313
Creating Software Metering Rules . . . 314
Software Metering Rule Matching . . . 315
Scheduling Data Flow . . . 316
Configuring Security Settings . . . 317
Adding and Deleting Software Metering Rules . . . 317
Enabling and Disabling Software Metering Rules . . . 318
Using Rules in Multitiered Hierarchies . . . 318
Software Metering Rules with the Same Name . . . 321
Using Software Metering with Terminal Services . . . 322
Using Software Metering Data . . . 323
Data Summarization . . . 324
Software Metering Reporting . . . 324
Software Metering Queries . . . 325

Scheduling Software Metering Maintenance Tasks . . . 326
Best Practices . . . 328
Distributing and Inventorying Programs to Be Monitored . . . 328
Configuring a Data Collection Schedule . . . 328
Configuring Software Metering Rules . . . 329
Addressing Privacy Concerns . . . 330
CHAPTER 9 Remote Tools . . . 331
SMS Remote Tools Overview . . . 332
Remote Assistance and Terminal Services Overview . . . 333
Installing, Enabling, and Configuring SMS Remote Tools . . . 335
Enabling and Configuring the SMS Remote Tools Client Agent on the SMS Site Server . . . 335
Installing SMS Remote Tools on Clients . . . 336
Installation on Clients Running Windows 2000 or Later . . . 337
Installation on Clients Running Windows NT 4.0 . . . 337
Preinstallation Testing for Clients Running Windows NT 4.0 or Later . . . 338
Installation on Clients Running Windows 98 . . . 339
Confirming SMS Remote Tools Installation . . . 339
Configuring Site-wide Settings . . . 340
Providing Remote Support . . . 345
Using SMS Remote Tools to Support Clients . . . 345
Establishing an SMS Remote Tools Connection . . . 346
Remotely Controlling Clients by Using SMS Remote Tools . . . 348
Conducting Two-Way Conversations with Client Users . . . 350
Diagnosing Client Hardware and Software Problems . . . 350
Testing Network Connectivity . . . 351
Running Commands and Programs on Remote Clients . . . 351
Transferring Files to and from Clients . . . 352
Restarting Remote Clients . . . 352
Using SMS Remote Tools at a Client . . . 353
Advanced Features of SMS Remote Tools . . . 354
Role of Wuser32.exe on Clients . . . 355
Client Security Settings . . . 356
Client Hardware Settings . . . 357

Video Acceleration . . . 359
Video Compression . . . 360
Video Acceleration on Clients Running Windows 2000 or Later . . . 361
Video Acceleration on Clients Running Windows NT 4.0 . . . 362
Improving the Performance of SMS Remote Tools . . . 367
CHAPTER 10 Maintaining and Monitoring the Network . . . 369
Using Network Monitor . . . 370
Capturing Network Traffic . . . 372
Examining Captured Data . . . 373
Using Network Monitor Experts . . . 373
Using SMS Network Diagnostic Tools on Remote Computers . . . 375
Capturing Traffic on Remote Computers . . . 376
Using Network Trace . . . 377
CHAPTER 11 Creating Reports . . . 379
Understanding Reporting . . . 380
Report Types . . . 381
Report Prompts . . . 381
Report Links . . . 382
Working with Reports . . . 384
Creating and Managing Reports . . . 385
Creating and Modifying SQL Statements . . . 404
Building an SQL Statement . . . 405
SQL Server Views . . . 409
Working with Dashboards . . . 415
Creating and Managing Dashboards . . . 415
PART 3 . . . 421
CHAPTER 12 Determining Product Compliance . . . 423
Using SMS for Product Compliance . . . 424
Compliance Analysis . . . 424
Compliance Solutions . . . 425
Viewing Product Compliance Data . . . 426
Customizing Product Compliance Data . . . 427
Customizing Product Compliance Data Manually . . . 427
Customizing Product Compliance Data Automatically . . . 429

CHAPTER 13 Maintaining and Monitoring SMS Systems 433
Maintenance and Monitoring Overview 434
Maintenance and Monitoring Plan 434
Maintenance and Monitoring Resources 434
Performance Monitor Counters 437
Using SMS Performance Monitor Counters 437
Maintenance Tasks 437
Predefined Site Maintenance Tasks 438
Custom Maintenance Tasks 443
Daily Tasks 444
Daily Site Maintenance Tasks 444
Daily Site Monitoring Tasks 444
Weekly Tasks 448
Weekly Site Maintenance Tasks 448
Weekly Site Monitoring Tasks 450
Periodic Tasks 451
Periodic Site Maintenance Tasks 451
Periodic Site Monitoring Tasks 454
Event-driven Maintenance Tasks 456
Maintenance Throughout the Hierarchy 458
Maintenance Operations 459
Attaching One Site to Another (Creating a Child Site) 460
Swapping the Computer of a Site Server 460
Rebuilding the Computer of a Remote SMS Site Database Server 461
Moving the SMS Site Database 462
Resetting a Site by Running SMS Site Reset 463
CHAPTER 14 Using the SMS Status System 465
Understanding Status Messages 466
Status Messages Defined 466
Status Message Characteristics 467
Other Message Characteristics 469
The Status Message Viewer 469
Interpreting System Status 471
Status Summarizer Concepts 472
Counts and States 472
Display Intervals 472

Status Indicators . . . 473
Thresholds . . . 474
Launching the Status Message Viewer and Other Tools . . . 474
Replication of Status Summaries Up the Site Hierarchy . . . 475
Monitoring and Troubleshooting with System Status . . . 476
Site Status . . . 477
Package Status . . . 484
Advertisement Status . . . 488
Configuring the SMS Status System . . . 489
Status Reporting Configuration . . . 490
Tuning Status Message Configuration with Status Filter Rules . . . 491
When to Use Status Filter Rules . . . 491
Configuring Status Filter Rules . . . 492
Sample Status Filter Rules . . . 496
Configuring Status Summarizers . . . 500
Deleting Status Messages . . . 500
Using the SMS Status System with the Windows Event Log . . . 501
CHAPTER 15 Backup and Recovery . . . 503
Planning for Backup and Recovery . . . 504
Preparing for Recovery . . . 504
Backing Up a Site . . . 508
The Backup SMS Site Server Task . . . 509
Backing Up a Site Using the Backup SMS Site Server Task . . . 513
Using SMSbkup.ctl to Control the Backup SMS Site Server Task . . . 515
Using AfterBackup.bat to Archive a Backup Snapshot . . . 522
Scheduling Considerations for the Backup SMS Site Server Task . . . 523
Enabling and Configuring the Backup SMS Site Server Task . . . 525
Verifying Success of the Backup SMS Site Server Task . . . 526
Backing Up a Secondary Site . . . 527
Backing Up the Central Site . . . 528
Monitoring Backup . . . 528
Using Third-Party Backup Tools to Back Up SMS Sites . . . 530
Recovering a Site . . . 530
Determining Whether a Site Recovery Operation Is Necessary . . . 531
Supported Configurations and Recovery Scenarios . . . 531
The Recovery Procedure . . . 532

Recovery and Repair Tools . . . 533
The Recovery Expert . . . 534
SMS Site Repair Wizard . . . 534
ACL Reset Tool . . . 537
Hierarchy Maintenance Tool . . . 538
Preparing for a Site Recovery Operation . . . 538
Data Traffic Issues . . . 539
Security Issues . . . 541
Managing the Site After Recovery . . . 542
APPENDICES . . . 545
APPENDIX A Using SMS to Distribute Office . . . 547
Overview of Office XP Deployment . . . 548
Office XP Operating System Requirements . . . 549
Important Concepts and Issues . . . 550
Package Definition Files . . . 551
System Files Update . . . 551
Multilingual User Interface Packs . . . 552
Windows Installer Versions . . . 552
Windows Installer Transform Files . . . 553
Windows Installer Patches . . . 553
How Office XP Uses Patches . . . 554
Using the Windows Installer Install on Demand Feature . . . 555
Windows NT 4.0 Low Rights Installation Issues . . . 556
Using the SMS Administrative Rights Installation Context . . . 556
Office Resource Kit Tools . . . 557
Office XP CD and Administrative Installation Source Issues . . . 558
Deploying Office XP in an Organization . . . 558
Business Requirements . . . 559
Enterprise Configuration . . . 559
Client Configuration . . . 560
Planning the Deployment . . . 560
Basic Planning Considerations . . . 561
Determine the Systems and Sites That Will Be Upgraded . . . 561
Determine SFU Requirements . . . 561
Plan for Clients Without Administrative Credentials . . . 562

Determine Which Clients Require Upgrades Prior to Installing Office XP . . . 563
Plan Installation Options . . . 564
Plan the Strategy for Collections and Program Advertisements . . . 564
Prepare and Customize the Office Source . . . 566
Deploying Office XP . . . 566
Maintaining and Updating Your Office XP Installation . . . 577
Distributing an Office XP Public Update . . . 577
Performing Administrative Patching of an Office XP Public Update . . . 578
Client Patching of an Office Public Update . . . 579
Distributing an Office XP Service Pack . . . 579
Updating Office XP Installation Settings . . . 580
Creating Updates Using the Custom Maintenance Wizard . . . 580
Applying the .cmw File to the Client . . . 580
Using Resilient Sources . . . 580
Frequently Asked Questions . . . 582
APPENDIX B Windows Management Instrumentation . . . 587
Introduction to WMI . . . 588
How SMS Uses WMI . . . 590
Understanding WMI . . . 591
WMI Architecture . . . 591
WMI Object Model . . . 593
WMI Schemas . . . 595
Comparing WMI to SQL Server . . . 597
WMI Browsing Tools . . . 598
CIM Studio . . . 598
WBEMTest.exe . . . 599
Visual Studio .NET . . . 600
WMI Command-line Tool . . . 600
Other WMI Browsing Tools . . . 601
Managing WMI . . . 601
Managing WMI Setup and Upgrade . . . 602
Using WMI Management Tools . . . 602
Backing Up WMI Data . . . 603
Understanding WMI Security . . . 604
Using MOF Files . . . 604

Troubleshooting WMI 606
WMI Troubleshooting Techniques 606
Verifying the State of the CIM Repository 608
Installation Issues 610
Connectivity Issues 610
Resource Consumption Issues 611
Programming Issues 611
Learning More About WMI 613
APPENDIX C Scripting SMS Operations 615
Understanding Scripting 617
Writing Scripts 618
Creating and Running a Simple Script 620
Developing Scripts 622
Scripting in Visual Basic 622
Connecting to WMI 623
Getting SMS Objects 624
Reporting Script 626
Displaying Distribution Point Status 628
Retrieving Lazy Properties 628
Advanced Queries 629
Working with SMS Objects 631
Collections 633
Collection Creation Example 634
Using Class-Specific Methods 636
Removing Rules from a Collection and Deleting Collections 637
Deleting Resources 637
Advertisements 638
Creating Advertisements 638
Modifying Advertisements 641
Unlocking Advertisements 642
Adding Assignments to an Advertisement 642
Packages 643
Creating Packages and Programs 643
Sending Packages to Distribution Points 644
Security Rights 646

Working with SMS Site Settings 647
Reporting Site Component Settings 649
Adjusting Component Settings 650
Setting the Site Comment for a Secondary Site 651
Embedding Properties 652
Creating Addresses 653
Adjusting Client Agent Settings 654
Adding Boundaries 656
Creating Site Systems 658
Managing Status Filter Rules 658
Scripting Console Operations 662
Scripting Client Operations 664
Creating DDRs for clients 665
Creating Status MIF Files 667
Scripting Advanced Client Operations 667
Debugging Scripts 669
Using Scripts on Web Pages 670
Understanding Support Implications of Scripted Solutions 671
Learning More 672
APPENDIX D Using SMS in International Organizations 675
Planning and Deploying Your Multilingual Site Hierarchy 676
Planning Multilingual Sites 676
Supported Localized Languages 677
Site Hierarchy Languages 679
Site Server Languages 680
Client Languages 684
International Client Pack 684
Multilingual Features 687
Local Language Display Configuration 688
SQL Server Configuration 689
Deploying Multilingual Sites 690
Sample Deployment Scenarios 690
Planning and Deploying International Client Packs 692

Planning ICP Deployments 693
ICP Design 693
ICP Testing 701
Deploying ICPs 704
ICP Installation 704
INDEX **711**

Getting Started

Welcome to Microsoft® Systems Management Server (SMS) 2003, a Windows-based product designed to make it easier for your organization to manage, support, and maintain a distributed network of computer resources.

The following sections will familiarize you with the wide range of technical information about SMS 2003. With this information, you can plan your SMS 2003 deployment, understand the features SMS 2003 offers, and how you can use those features to benefit your organization.

Technical Resources

SMS 2003 includes comprehensive product documentation and other technical resources that help you deploy and use SMS.

Online Library

All the information you need for deploying and using SMS 2003 is provided in the SMS Online Library. The Online Library includes the following:

- An electronic version of the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.
- Information about how to order printed books for SMS, including the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* and the *Microsoft Systems Management Server 2003 Operations Guide.*
- Information about where to find electronic versions of the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* and the *Microsoft Systems Management Server Operations Guide*.
- SMS Help, which provides information about how to use the SMS Administrator console to manage your sites.

- Release Notes, which contain critical information about SMS.
- Links to the SMS Web site at http://www.microsoft.com/smserver/default.asp. On this site, you can find SMS-related information, such as technical papers, product news, and software updates. The SMS Web site also provides specific information about how to use SMS with other Microsoft products, such as Microsoft Windows® XP and Office XP.

Running the SMS Online Library

- From the **Start** menu, click **Programs**, click **Systems Management Server**, and then click **SMS Online Library**.

  – Or –

  Right-click **SMS Online Library** in the SMS Administrator console tree and click **Run Online Library**.

Product Documentation Available for SMS

Before you start using SMS 2003, you should read the following books to become familiar with the product.

Table A.1 SMS 2003 Books

Book	Description
Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	This book contains valuable information about planning for deploying SMS in your organization, important concepts of SMS, and directions for installing SMS and upgrading from previous versions. This book is key to understanding SMS.
Microsoft Systems Management Server 2003 Operations Guide	This book provides information about configuring and using SMS.

These books are available in several different formats:

- Help on the product CD (*Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* only)
- .Pdf files can be downloaded from the Web
- Searchable content on Microsoft TechNet

For more information about accessing these resources, see the information about the Online Library in the previous section.

Help is also provided for all SMS features, including the SMS Administrator console. To access Administrator Help in the SMS 2003 Administrator console, press **F1,** or right-click any item and select **Help** from the pop-up menu.

Keeping Your Technical Knowledge Current

To help you stay current with the latest information about SMS 2003, the SMS product documentation and other helpful resources will be updated on a regular basis on the Web after the initial release of SMS 2003. For example, you'll be able to download updated troubleshooting information from the SMS Web site that reflects new knowledge of the product gained through real-world experience since the product's initial release.

You should regularly check the SMS Web site at http://www.microsoft.com/smserver/techinfo/default.asp for updates to important technical references and product documentation that help you stay informed about SMS.

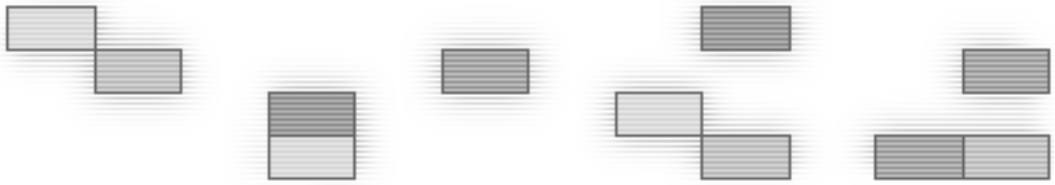
Document Conventions

The following conventional terms, text formats, and symbols are used throughout this book.

Convention	Description
Bold	Indicates the actual commands, words, or characters that you type in a dialog box or at the command prompt. Also indicates named user interface elements (**Program Properties** dialog box, for example.)
Italic	Indicates a placeholder for information or parameters that you must provide. For example, if the procedure asks you to type *filename*, you must type the actual name of a file. An italic typeface also indicates new terms and the titles of other resources in the Systems Management Server documentation set.
ALL UPPERCASE	Indicates an acronym, key, or macro name. You can use lowercase letters when you type directory names or filenames in a dialog box or at the command prompt indicated.
`Monospace`	Represents examples of screen text or entries that you might type at the command line or in initialization files.
►	Indicates a procedure.
▪	Indicates an unordered list of related information (not a procedure).

PART 1

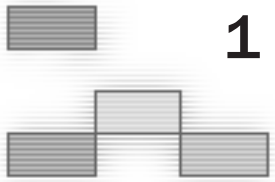
Deploying SMS



This part of the *Microsoft Systems Management Server 2003 Operations Guide* introduces in-depth technical information that will enhance your ability to use specific Systems Management Server 2003 features.

CHAPTER 1

Scenarios and Procedures for Deploying SMS 2003



This chapter builds on the deployment planning information in the *Microsoft® Systems Management Server 2003 Concepts, Planning, and Deployment Guide*. Each step in the deployment scenarios presented in this chapter will refer you to existing documentation for a more detailed discussion of the issues and concepts related to that step. When needed, additional information is provided for that step in this chapter. Although it is not essential that you have already read the existing documentation contained in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*, it is strongly recommended that you do so to enhance your understanding of the material contained in this chapter. It is important that you spend an appropriate amount of time and resource planning and designing your Systems Management Server (SMS) 2003 sites and hierarchy.

In This Chapter

- Overview of the Deployment Process
- Part 1: Hierarchy-Specific Questions
- Part 2: Site-Specific Questions
- Part 3: SMS 2003 Deployment Scenarios
- Post-installation Considerations

Overview of the Deployment Process

The *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* stresses the importance of developing a thorough and complete strategy for deploying SMS 2003 in your organization. The information in this chapter can facilitate the development of such a strategy. The scenarios in this chapter are meant to be adaptable to the unique needs of your organization instead of being a prescribed method that fits every organizational model. You should use the scenarios in this chapter as guidelines for developing your own implementation strategy. For example, you might choose to implement a side-by-side upgrade of SMS 2003 at the central site level, and implement in-place upgrades at specific child sites.

This chapter provides you with a roadmap for developing a deployment plan for your SMS 2003 sites by offering a prescriptive guide using a flowchart model built around three principal deployment scenarios. In addition, this chapter directs you to the relevant conceptual, planning, and operational material that exists in other SMS 2003 documentation. Each flowchart includes action items for you, such as reading a specific resource topic or carrying out a task. The deployment scenarios are designed to be flexible. Except for certain explicit cases, you can apply them to any portion of your SMS hierarchy in addition to the hierarchy as a whole.

The three principal deployment scenarios are:

- New deployment of SMS 2003
- In-place upgrade of SMS 2003
- Side-by-side upgrade of SMS 2003

New deployment of SMS 2003 This scenario represents a fresh installation of SMS 2003 in an organization where no previous SMS installation exists, or where you plan to remove any previous installations of SMS. In this scenario, you do not need to consider any existing SMS 2.0 hierarchy and can develop and implement a new SMS 2003 site hierarchy. It is still important to properly evaluate the existing environment and design the SMS hierarchy appropriately.

In-place upgrade of SMS 2003 This scenario represents an upgrade of an existing SMS 2.0 hierarchy. In this scenario, you plan to maintain the existing SMS hierarchy, the existing CAP and distribution point roles, and the existing SMS site boundaries. SMS clients remain assigned to their current SMS sites. In this scenario, you need to consider whether a new SMS 2003 site can manage your current SMS client computers. It might be that SMS 2003 cannot support some of your existing client computers. In this case, you might choose to maintain an SMS 2.0 site indefinitely — called a holding site — to support those clients. Consequently, you need also to be aware of any interoperability issues between the SMS 2.0 site and the SMS 2003 site that can affect your SMS hierarchy. Holding sites and interoperability issues are described later in this chapter.

Side-by-side upgrade of SMS 2003 This scenario represents an implementation of a new SMS 2003 hierarchy that you plan to migrate existing SMS clients to. You can choose to implement a side-by-side upgrade to:

- Use new or updated server hardware.
- Reflect changes made in your organizational structure.
- Compartmentalize the usage of different SMS 2003 features, for example, managing mobile clients in an SMS site separate from that which is managing desktop clients.
- Take advantage of the increased scalability of SMS 2003 Advanced Client and reduce the overall number of SMS sites in your hierarchy.
- Maintain a functioning SMS site and managed clients while rolling out a new SMS infrastructure.

Client Support

This chapter categorizes SMS clients into three classes to distinguish how SMS supports them. Table 1.1 describes the type of client maintained in each class. Table 1.2 describes the Microsoft Windows® operating systems supported by clients in each class.

Table 1.1 SMS Client Classes

Class	Description
Class A	Supported by SMS 2003 sites. Clients in this class generally run SMS 2003 Advanced Client, but can also run the SMS 2003 Legacy Client, and the SMS 2.0 client.
Class B	Supported by SMS 2003 sites, but the client operating systems do not run the SMS 2003 Advanced Client. Clients in this class generally run the SMS 2003 Legacy Client, but can also run the SMS 2.0 client.
Class C	Supported only by SMS 2.0 sites. Clients in this class run the SMS 2.0 client.

Table 1.2 Windows Operating Systems Supported by Each SMS Client Class

Operating system	Class A	Class B	Class C
Windows Server™ 2003 family	X		
Windows 2000 family	X		
Windows XP Professional	X		
Windows XP Home	N/A	N/A	N/A
Windows NT® 4.0 Service Pack 6 (with Internet Explorer 5.0 or later)		X	

(continued)

Table 1.2 Windows Operating Systems Supported by Each SMS Client Class *(continued)*

Operating system	Class A	Class B	Class C
Windows NT 4.0 Service Pack 5 and earlier			X
Windows Millennium Edition			X
Windows 98 (with Internet Explorer 5.0 or later)		X	
Windows 98			X
Windows 95			X

Class C computers are not capable of supporting either the Legacy Client or the Advanced Client because of operating system incompatibility. If SMS 2.0 sites currently manage these clients, you must decide whether you need to continue supporting these clients. If so, then you need to manage them with an SMS 2.0 site until you can upgrade them to either the Legacy or Advanced Client, or no longer need to maintain them as SMS clients. This kind of SMS 2.0 site is known as a holding site.

Holding site SMS installs client software for Class A and Class B clients according to the methods outlined in Chapter 17, "Discovering Resources and Deploying Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*. Because SMS 2003 sites do not support Class C computers, SMS 2003 does not install any SMS client software on Class C computers. If Class C computers previously were SMS 2.0 site clients, they effectively become orphaned clients in an SMS 2003 site.

A holding site is a designated SMS 2.0 site in the SMS 2003 site hierarchy that manages Class C computers. The holding site is a child site of an SMS 2003 site. The site boundaries of the holding site overlap with those of the SMS 2003 site or sites that have Class C computers.

For those computers that reside in the overlapping boundaries of SMS 2.0 and SMS 2003 sites, SMS determines which client type to install according to the Logon Script-initiated Client Installation command (Capinst.exe) and the computer's operating system. In this case, Class C clients automatically become clients of the SMS 2.0 holding site rather than becoming orphaned.

Your decision to install the SMS 2003 Advanced Client or the SMS 2003 Legacy Client — supported by Class A and Class B computers — depends on more than the supported operating system.

Resources

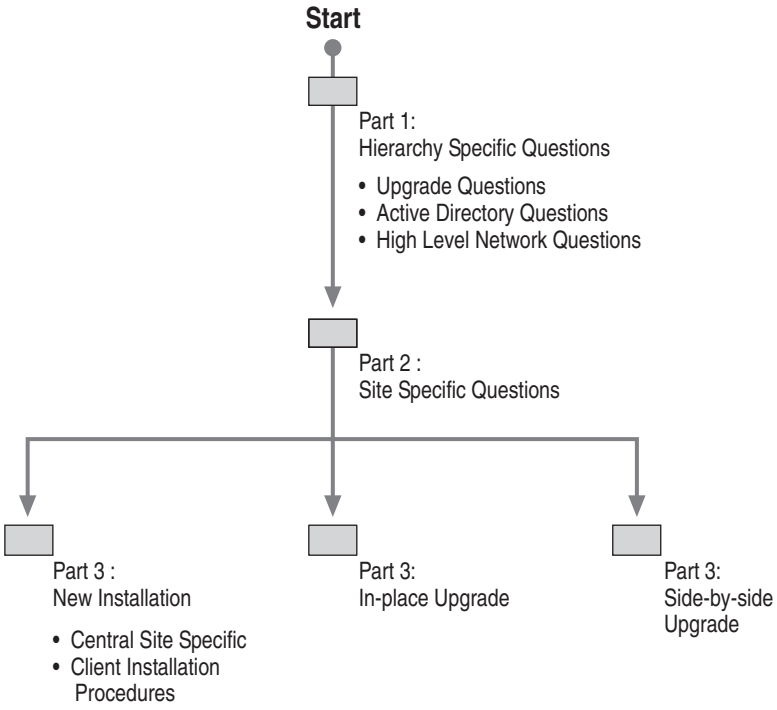
Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For more information about the distinction between SMS 2003 client types:	
Chapter 4	**Entire chapter recommended**
For more information about the interoperability between SMS 2003 and SMS 2.0 sites and the effect on clients:	
Chapter 11	**Entire chapter recommended**
For more information about planning your client deployment:	
Chapter 10	**Entire chapter recommended**

SMS Deployment Components

There are three main components to consider as you deploy SMS 2003 in your organization. Figure 1.1 shows each component, labeled Part 1, Part 2, and Part 3, and how that component fits into the deployment process along with the high-level steps you should follow when implementing your deployment plans.

- Part 1 describes deployment questions that are specific to planning your SMS hierarchy.
- Part 2 describes deployment questions that are specific to planning each site in your SMS hierarchy.
- Part 3 describes each of the three deployment scenarios you might choose.

Figure 1.1 Main components of the SMS 2003 site deployment process



Part 1

This part of the deployment process outlines hierarchy-specific questions for your consideration, including the following:

- Do you have an existing SMS 2.0 site?
- Do you plan to upgrade your existing site?
- Is Active Directory® implemented in your environment?
- How does your network infrastructure relate to the location of servers and user computers?

Part 2

This part of the deployment process follows Part 1 and outlines site-specific questions for your consideration, including the following:

- Are you implementing a central site or a child site?
- How many clients are reporting to the SMS site?
- What client types do you need to manage?
- What client installation methods do you plan to use?

Part 3

This part of the deployment process follows Part 2. The answers to the questions posed in Parts 1 and 2 determine which of the three SMS 2003 deployment scenarios you might implement, and the steps required for each scenario.

New installation

- Are you managing Advanced Clients at this site?
- Are you managing Legacy Clients at this site?
- Are you configuring roaming boundaries?
- What client installation methods are you using?

In-place upgrade

- What are the results of running the Deployment Readiness Wizard?
- Do you need to migrate an existing custom SMS_def.mof file?
- Do you require a holding site?
- Do you plan to consolidate your existing SMS site infrastructure?

Side-by-side upgrade

- Are you installing a new SMS central site?
- Are you implementing roaming boundaries?
- What client installation methods are you using?

Each part and scenario is described more fully in subsequent sections of this chapter.

Part 1: Hierarchy-Specific Questions

This section provides a pre-deployment checklist of questions to ask and steps to perform that help you determine the type of deployment scenario to implement in your organization. The section uses four flowcharts to guide you through the questions and help you determine which of the three deployment scenarios is appropriate for your organization.

Before you begin planning your deployment, it is recommended that you read the chapters referenced in Resources 1 relating to background concepts in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*. These chapters provide the detailed information you need about the various parts of an SMS 2003 site, and issues you must consider before you deploy SMS.

Resources 1

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For more information about SMS sites, and how they are attached to build an SMS hierarchy:	
Chapter 2	**Entire chapter recommended**
For more information about how core features of SMS work, how you can use each of those features to benefit your organization, and how these features are integrated to perform common tasks in an organization:	
Chapter 3	**Entire chapter recommended**
For more information about the SMS client, and the client discovery and installation methods provided by SMS:	
Chapter 4	**Entire chapter recommended**
For more information about SMS security features, including security modes, accounts and groups, and object-level security:	
Chapter 5	**Entire chapter recommended**

This section contains the following topics:

- Upgrade Questions
- Active Directory Questions
- Network Questions

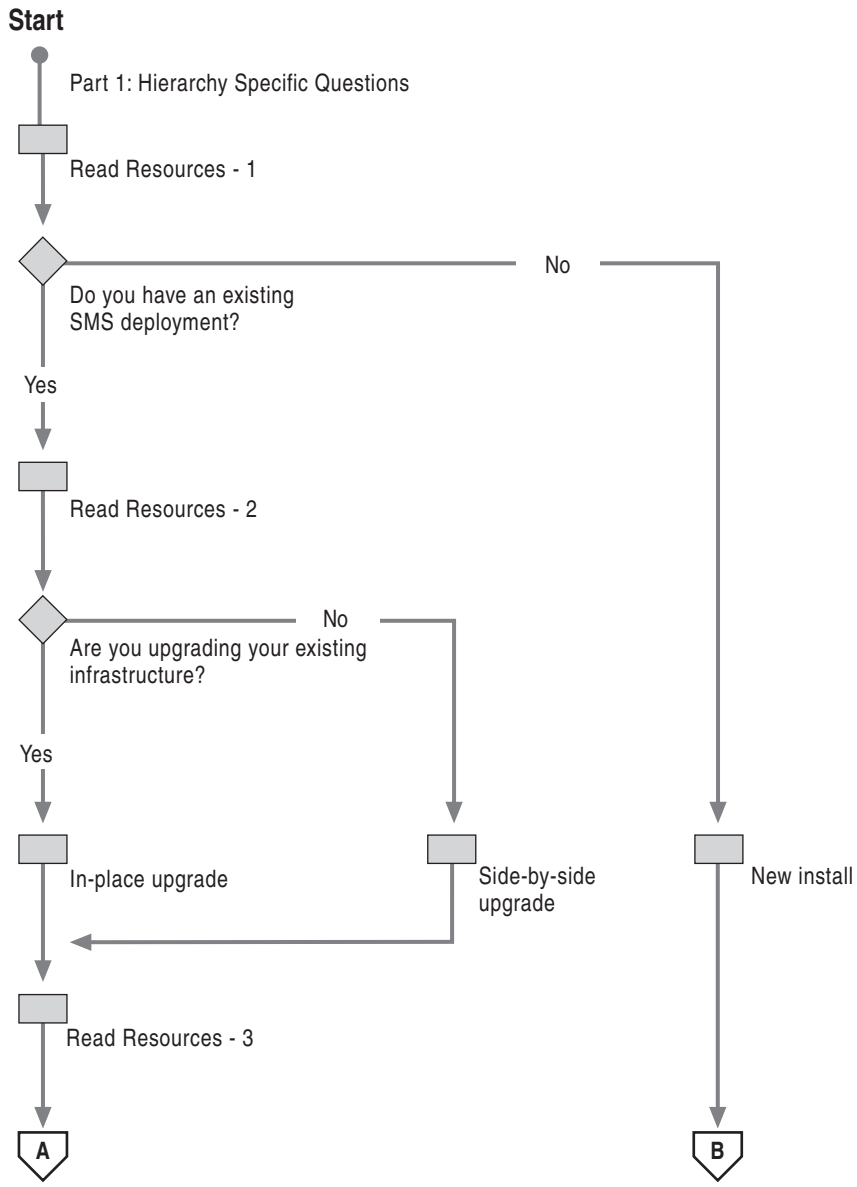
Upgrade Questions

The first flowchart, shown in Figure 1.2, lists questions to ask that help you determine whether you need to upgrade an existing installation of SMS, and what kind of installation is appropriate.

Note

All down arrows in each flowchart represent a positive response to a question box. All right arrows represent a negative response to a question box.

Figure 1.2 Upgrade questions flowchart



Do you have an existing SMS deployment?

The first question to consider as you plan your SMS 2003 deployment is whether you have an existing SMS deployment in your organization. If you do not have an existing SMS installation, then you are deploying SMS 2003 as a new installation. In this case, see the "Active Directory Questions" section later in this chapter.

You can also choose to remove your existing SMS installation altogether. In this case, remove SMS first, and then see the "Active Directory Questions" section later in this chapter. See the documentation for your previous version of SMS for details about how to remove SMS.

If you choose to remove SMS and your SMS hierarchy consists of several SMS sites, you must remove SMS from every site. It is recommended that you begin with the lowest level sites in the hierarchy first, ending with the central site. At a minimum, you need to have performed the following steps:

- Remove the SMS site from the existing hierarchy.
- Remove all clients that are assigned to the SMS site.
- Remove all client software from client computers.
- Remove all SMS site system roles from servers.
- Remove SMS site server software by running SMS Setup.
- Remove all SMS-specific registry keys from the SMS site server. For more information, see article 217044 in the Microsoft Knowledge Base at http://support.microsoft.com.
- Remove all SMS-specific accounts from the local SMS site server and from the site's Windows domain unless you want to reuse those accounts for the new SMS 2003 site.

One way that you can remove all clients assigned to a site in addition to all client software from client computers is to remove all site boundaries, and then wait one day (23 hours) for the clients to initiate the uninstall process.

Note

You must account for clients that are offline when you remove the site boundaries. These will not begin the uninstall process until they are online again.

If you have an existing installation of SMS, and you plan to migrate SMS clients from the existing installation to SMS 2003, you must familiarize yourself with the relevant interoperability considerations related to SMS 2.0 and SMS 2003 sites, and with planning issues relating to an upgrade from SMS 2.0 to SMS 2003.

Resources 2

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For a detailed discussion of interoperability issues with SMS 2.0:	
Chapter 6	**Interoperability of SMS 2.0 Features with SMS 2003 Features**
For a detailed discussion of general planning issues related to upgrading from SMS 2.0:	
Chapter 11	**Entire chapter recommended**

Are you upgrading your existing infrastructure?

This question has two considerations. You must consider whether to use the existing SMS site infrastructure or whether you intend to modify the number and assignment of site system roles. Site system roles include client access point (CAP), distribution point, management point, server locator point, reporting point, and site server.

You also need to decide whether you want to use your existing server hardware to support SMS 2003, or whether you want to use new hardware. If you choose to use the existing hardware, you are performing an in-place upgrade.

If your existing SMS hierarchy consists of many SMS sites, consider whether you should consolidate those sites. It might be appropriate to develop a new design for your SMS hierarchy. You might also consider upgrading your existing hardware or using new hardware to support your SMS servers. If you plan to use new hardware, consolidate your existing site, or design a new site hierarchy as part of your upgrade strategy, you might be performing an in-place upgrade or a side-by side upgrade.

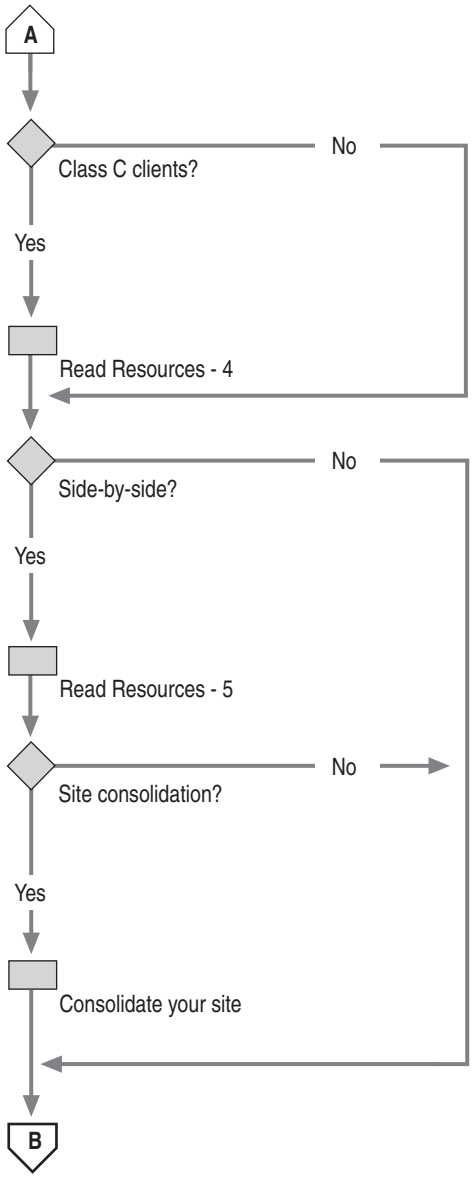
Resources 3

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about how to design your site and plan your hardware choices:	
Chapter 7	**Entire chapter recommended**
Chapter 8	**Entire chapter recommended**
Chapter 9	**Entire chapter recommended**
Chapter 11	**Entire chapter recommended**

Options for Client Migration

The flowchart in Figure 1.3 lists the questions that determine what options you have for client migration for the in-place and side-by-side migration scenarios.

Figure 1.3 Options for client migration flowchart



For both in-place and side-by-side deployment scenarios, if you have clients that are in the Class C category described in the Client Support topic earlier, you must decide whether you want to continue managing these clients with SMS. If so, then you need to implement a holding site for those clients. If not, then remove the SMS client software from those clients so that they do not become orphaned. Clients that are in the Class A and Class B categories become members of the SMS 2003 site according to the client installation method you select for the site, and the site boundaries and roaming boundaries you configure.

Resources 4

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For a detailed discussion about holding sites:	
Chapter 11	**In-Place Hierarchy Upgrades** **Example Scenario 1** **Example Scenario 2** **Deciding When to Upgrade a Flat Hierarchy**
For a detailed discussion of client installation methods:	
Chapter 17	**Installing the Advanced Client** **Installing the Legacy Client**
For detailed information about configuring SMS site boundaries:	
Chapter 10	**Configuring Site Boundaries and Roaming Boundaries**
For detailed information about how to configure logon scripts to separate Class C from Class A and B computers during logon script initiated installation:	
Chapter 6	**Client Discovery and Installation**

In the case of a side-by-side migration, you should understand the extra scalability you get by using the Advanced Client. This does not mean that for Advanced Clients, different site systems can be on different networks. An SMS site still must be well connected.

Resources 5

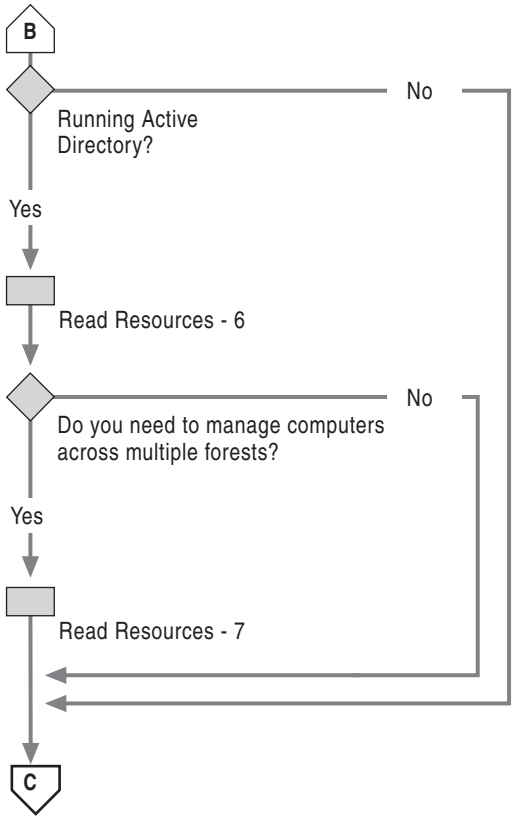
Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about altering your hierarchy as you upgrade, and the performance advantages you get from using the Advanced Client:	
Chapter 11	**Side-By-Side Hierarchy Upgrades**
Chapter 9	**Entire chapter recommended**

If you plan to consolidate your SMS site as part of a side-by-side migration, the next step is to do the consolidation. In this case, add the boundaries of old SMS sites to the boundaries of the consolidated site. Use SMSMan.exe with the **/F** switch or referencing a script to assign computers to the consolidated site. When you finish assigning the computers to the consolidated site, remove SMS software from the old SMS sites.

Active Directory Questions

The flowchart in Figure 1.4 lists the questions to consider when you are deploying SMS in an Active Directory environment.

Figure 1.4 Active Directory questions flowchart



In the case of all three deployment scenarios, if you are implementing SMS 2003 in an Active Directory environment, you have the benefit of implementing advanced security, the preferred security mode. You must understand how SMS 2003 uses Active Directory and know the requirements for using advanced security. In particular, you should understand how to extend the Active Directory schema for SMS, how to use Active Directory site names for your SMS site boundaries and roaming boundaries, and how to manage SMS clients that roam from SMS site to SMS site. Extending the Active Directory schema is a forest-wide action. If you extend the schema for one SMS site in the forest, the schema is extended for use by all SMS sites in the forest.

Resources 6

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about extending the Active Directory schema:	
Chapter 10	**Active Directory Considerations**
Chapter 15	**Extending the Active Directory Schema**
For detailed information about configuring Active Directory site boundaries and client roaming:	
Chapter 2	**Site Boundaries** **Roaming and Roaming Boundaries**

If you need to use SMS across multiple forests, there are several issues for you to consider. Be aware that a single SMS site cannot span multiple Active Directory forests, although it can span multiple domains within a single forest. Also, all SMS site systems must be in the same Active Directory forest as the SMS site server. There are also considerations across forests in the following areas:

- Site-to-site communications
- Client communications
- Secure key exchange
- Client global roaming

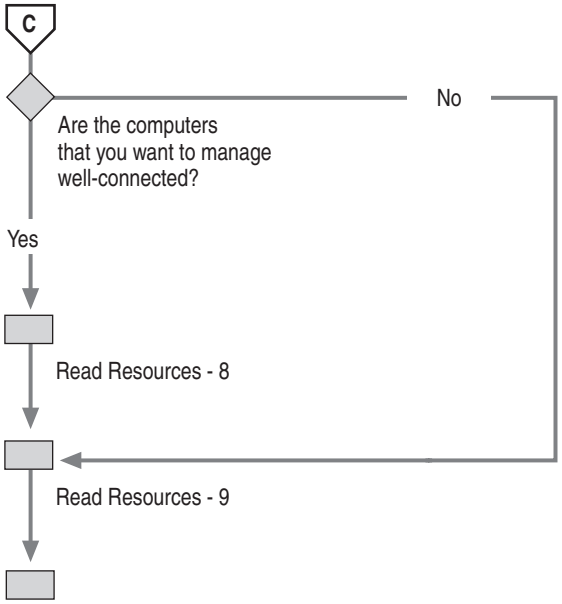
Resources 7

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about supporting SMS 2003 across multiple forests:	
Chapter 8	**Active Directory Considerations**

Network Questions

The flowchart in Figure 1.5 lists the questions to consider when you are deploying SMS that are specific to your network infrastructure.

Figure 1.5 Network questions flowchart



You need to consider your network infrastructure when designing your SMS site and hierarchy. Some SMS site tasks can consume considerable bandwidth. It is also recommended that SMS site systems and SMS clients be well-connected. The speed and bandwidth usage of your network is a significant consideration when deploying your SMS site. The resources described in Resources 8 help you to determine speed and bandwidth usage and whether your SMS site systems and SMS clients are well-connected.

It is important that you plan for the appropriate number of SMS sites and site systems that your network can accommodate. You might also consider upgrading or reconfiguring your network infrastructure as well.

Resources 8

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For information about network considerations when planning your SMS site:	
Chapter 7	**Analyze Your Environment**
For information about how to determine the appropriate number of sites:	
Chapter 8	**Business Considerations**

Resources 9

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For information about network boundaries for SMS sites:	
Chapter 2	**Site Boundaries** **Roaming and Roaming Boundaries**
Chapter 8	**Technical Considerations** **Planning Site Boundaries and Roaming Boundaries**
For information about capacity planning issues to consider that are related to the network:	
Chapter 9	**Network Considerations**

Part 2: Site-Specific Questions

This section continues the process begun in Part 1. This section provides a pre-deployment checklist of questions to ask that are specific to the SMS site you are implementing. This section uses two flowcharts to guide you through the questions and help you determine how to configure your SMS site. As with the flowcharts shown in Part 1, you can use these flowcharts to plan the deployment or upgrade of each site in your hierarchy.

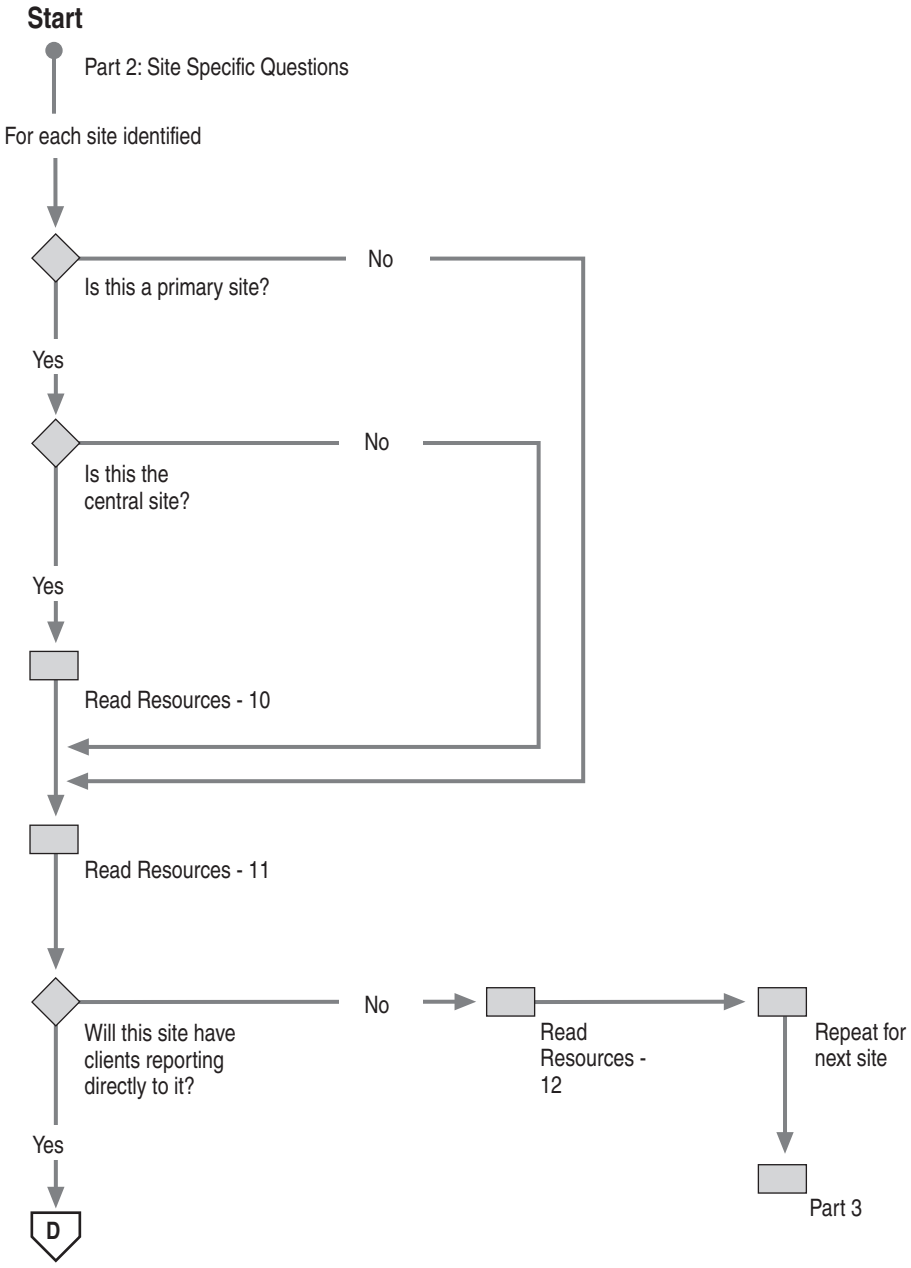
This section contains the following topics:

- Site Configuration Questions
- Client Configuration Questions

Site Configuration Questions

The flowchart in Figure 1.6, lists the questions that determine what type of SMS site to install, and the issues to consider for each type.

Figure 1.6 Site configuration questions — choosing a site



Based on your answers to the questions listed in Part 1, you determine the number of SMS sites and their configuration. You then decide whether the SMS site is a primary site or a secondary site. The resources listed in Resources 1 help you to make this determination.

The topmost SMS site in your SMS hierarchy is the central site. The SMS central site is always an SMS primary site. There are issues for you to consider that are specific to the SMS central site. The SMS site database at the central site stores aggregate inventory and software metering data and status from the SMS hierarchy, and collects details about any collections, packages, or advertisements created at the central site. At the central site, you can view and manage all sites and computers in the SMS hierarchy. Because all status and client data flows up in the hierarchy to the central site, adding a large number of clients to this site can diminish central site server performance and client performance. Consequently, especially in large organizations, the central site should not manage clients.

The SMS central site generally maintains the server locator point for the SMS hierarchy. Because the SMS central site database contains data from other SMS sites below it in the SMS hierarchy, you might install the reporting point site system on the central site server.

Each primary site you deploy, including the central site, uses a site database to hold the data collected from the site. Management points, server locator points, and reporting points also use the SMS site database. See the "Getting Started" chapter in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* for a complete list of requirements for the SMS site database.

On the Windows Server 2003 family of servers, the following components used by certain SMS 2003 site systems are not enabled by default:

- Background Intelligent Transfer Service (BITS)
- Internet Information Services (IIS)
- Web Distributed Authoring and Versioning (WebDAV) extensions for IIS
- Active Server Pages (ASP)

If you are deploying SMS 2003 site systems to Windows Server 2003 servers, you must enable the appropriate component for the appropriate SMS site system. Table 1.3 describes which of these components you must enable for each SMS site system.

Table 1.3 Windows Server 2003 Components to Enable for SMS 2003 Site Systems

SMS site system	Windows Server 2003 component to enable
Distribution point	Enable IIS Enable WebDAV extensions for IIS
Management point	Enable IIS Enable BITS
Reporting point	Enable IIS Enable ASP
Server locator point	Enable IIS

For a primary site and a secondary site, you need to decide which security mode to run: advanced security or standard security. Advanced security is the preferred mode because it takes advantage of local system and computer accounts that are automatically maintained by the operating system. For example, SMS runs its server components in the local system security context, or using the computer account instead of a user account. Also, SMS parent and child site servers running advanced security can use each other's computer account to send information to back and forth. Standard security requires more user accounts to manage the same processes.

If the SMS site is managing clients, there are client-specific issues to consider when choosing the appropriate security mode. For example, if you plan to use Legacy Clients in your advanced security SMS site, you must create at least one SMS Client Connection Account before installing the Legacy Clients. Advanced Clients might require the Advanced Client Network Access Account when an advertised program needs to access a share on a server other than the distribution point or when the distribution point or content server is in a Windows NT 4.0 domain or in another forest.

Resources 10

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about the role of a primary site and the central site, and considerations for configuring site systems for the central site:	
Chapter 8	**Determining the Locations and Types of Site Servers** **Advantages of Multiple Sites**
Chapter 10	**Deploying Central and Administrative Sites**

Resources 11

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about the SMS site database, and considerations for planning for and configuring the SMS site database:	
Chapter 10	**SMS Site Database Server Considerations** **Preparing Site System Computers**
For detailed information about capacity planning considerations related to the SMS site database:	
Chapter 9	**Modeling Principles for Sizing and Capacity Planning** **Server Activities** **Estimating the Number of Clients and Objects** **Determining SMS Site Database Server Requirements**

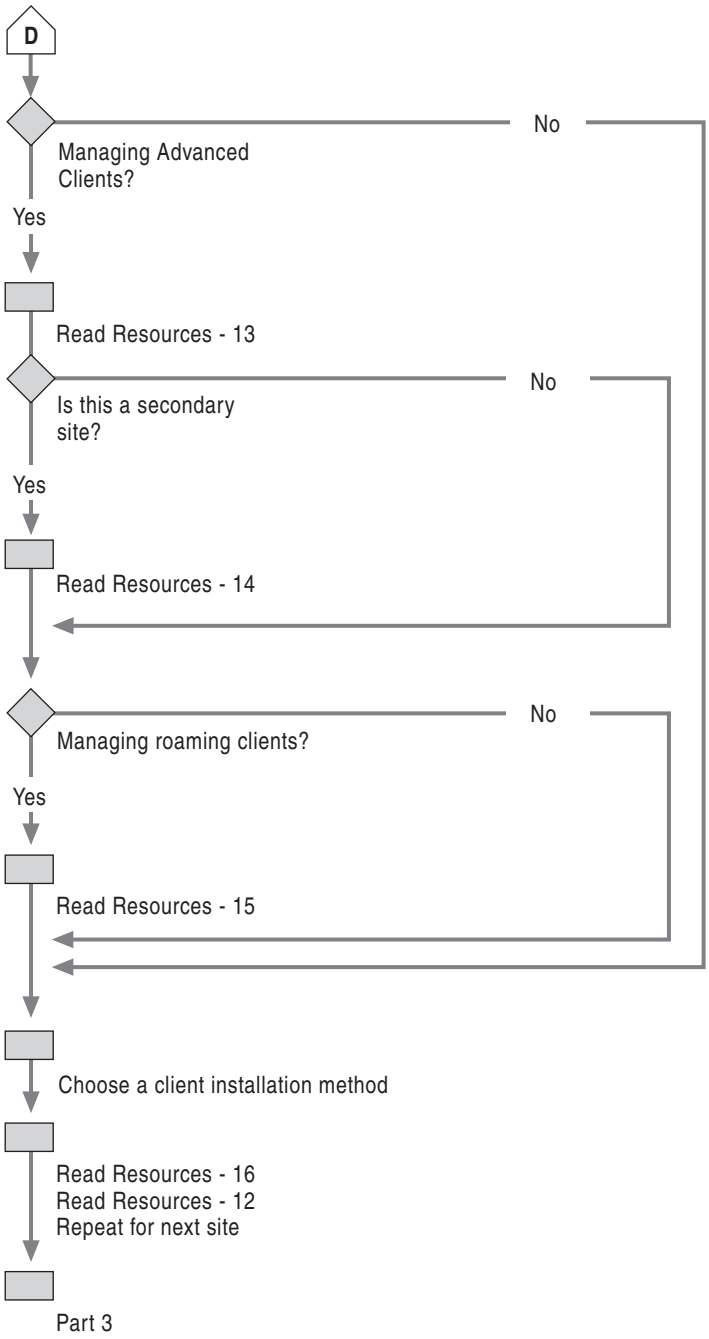
Resources 12

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about Advanced and Standard security, and the affect each mode has on the SMS site and SMS clients:	
Chapter 5	**SMS Security Modes**
Chapter 8	**Active Directory Considerations** **Primary and Secondary Site Decisions**
Chapter 12	**Security Considerations for Site and Hierarchy Design** **Tightening SMS Security**

Client Configuration Questions

The flowchart in Figure 1.7 lists the questions that determine what type of SMS clients you are installing in your SMS site, and the issues to consider for each type of client.

Figure 1.7 Site configuration questions — choosing a client



If the SMS site manages client computers, you need to determine whether the SMS site manages Advanced Clients, Legacy Clients, or both. Each client type has its own considerations. For example, Advanced Clients use the management point to obtain Advanced Client policy and configuration information, and to send client data to the SMS site database. Legacy Clients use the CAP to obtain configuration information and send client data to the SMS site database.

Because the Legacy Client is based on the earlier technology of the SMS 2.0 client, it relies heavily on domain accounts to carry out key tasks on the SMS client computer such as installing software in an administrative context when the logged-on user account does not have the appropriate security credentials. The Advanced Client, though, is engineered to use the local system security context and the computer account to carry out these same key tasks, making the Advanced Client a much more secure. It is strongly recommended that you install the Advanced Client as the preferred client on all your SMS client computers running the Windows 2000 or later operating system.

WARNING

Microsoft currently plans to discontinue support for the SMS Legacy Client on computers running the Windows 2000 or later operating system platforms with the release of SMS 2003 SP1.

Although Advanced Clients are only assigned to primary sites, you can install management points on both primary and secondary sites. A management point on a secondary site is known as a proxy management point. It is used for roaming Advanced Clients if roaming boundaries are enabled for the primary site. When you install an SMS 2003 secondary site, and that secondary site does not have a proxy management point installed, the secondary site's boundaries are added to the roaming boundaries of the primary site. An SMS 2.0 secondary site's boundaries are also added to the roaming boundaries of the parent site. However, if an SMS 2003 secondary site has a proxy management point installed, that secondary site's boundaries are not added to the roaming boundaries of the primary site.

Advanced Clients located at a secondary site and reporting to a management point at a parent primary site across a WAN link might have an effect on the available bandwidth of the WAN link between the secondary site and its parent primary site. Significant network traffic can be produced when client status and hardware or software inventory data is sent to the parent primary site. Because an Advanced Client can be assigned only to a primary site, network traffic generated by Advanced Client policy requests also reduces the available bandwidth between the two sites. Proxy management points increase bandwidth efficiency by servicing roaming clients that are within the secondary site's roaming boundaries. You need to determine whether your Advanced Clients can benefit from a proxy management point in an SMS secondary site.

Resources 13

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about the Advanced and Legacy Client types:	
Chapter 4	**SMS Clients**

Resources 14

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about CAPs, management points, proxy management points, and their role in the SMS hierarchy:	
Chapter 8	**Planning Site Boundaries and Roaming Boundaries**
For considerations related to capacity planning for CAPs and management points:	
Chapter 9	**Sizing SMS Component Servers**

Resources 15

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about managing roaming clients:	
Chapter 2	**Roaming and Roaming Boundaries**

You need to select an installation technique for installing the SMS client software on computers that the SMS site manages. SMS client installation techniques include:

- Using the Client Push Installation method in the SMS 2003 Administrator console.
- Initiating a program file at the client to install the client software, as follows:
  - Logon Script-initiated Client Installation.
  - Manually running a program file.
  - Using Windows Group Policy.
  - Using SMS software distribution or some other software distribution mechanism to advertise and run a program file.
- Installing the Advanced Client on a computer master image, and imaging that computer to other computers.

Resources 16

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about each client installation technique:	
Chapter 10	**Client Deployment Planning**
Chapter 17	**Installing and Configuring SMS Clients**
For detailed information about SMS accounts required for client installation:	
Chapter 5	**SMS Accounts and Groups**
Chapter 12	**Planning SMS Accounts**
Chapter 17	**Installing and Configuring SMS Clients**

For a primary site and a secondary site, you need to decide which security mode to run: advanced security or standard security. For more information, see the "Site Configuration Questions" section earlier in this chapter.

Part 3: SMS 2003 Deployment Scenarios

This section describes three deployment scenarios that you might choose as you define your SMS 2003 deployment strategy. The three scenarios are most effective if you complete the hierarchy-specific and site-specific questions and tasks described earlier in this chapter.

The three scenarios described in this section are not the only deployment methods that you might implement. You might apply a different scenario to each SMS site within your SMS hierarchy depending on the requirements of each site. The unique needs of a specific site might require you to modify the deployment steps appropriately. These three scenarios are meant to be helpful guides instead of rigid rules.

You must consider the effect that the deployment method will have on your organization. For example, you might intend to upgrade an existing SMS 2.0 site to SMS 2003 using the existing SMS servers and site system roles. This case implies that an in-place upgrade is appropriate. However, during the course of the in-place upgrade, some existing SMS clients might be left unmanaged and Class C clients can become orphaned. At the same time, your organization's service level agreements (SLAs) regarding the management of client computers might require that SMS clients must always be managed. Furthermore, you might not be able to suspend those SLAs. Given these considerations, a side-by-side upgrade might be the better choice of deployment method.

This section contains the following topics:

- New Installation
- In-Place Upgrade
- Side-by Side Upgrade

Some of the steps described in the following sections pertain to one or more scenarios. Instead of repeating these steps for each scenario, the flowcharts associated with each scenario identify which flowcharts refer to a specific set of steps. For example, each scenario refers to the installation of management points. When you get to that point in the flowchart for each scenario, the scenario flowchart indicates that you should refer to the management point installation flowchart for steps specific to the installation of a management point.

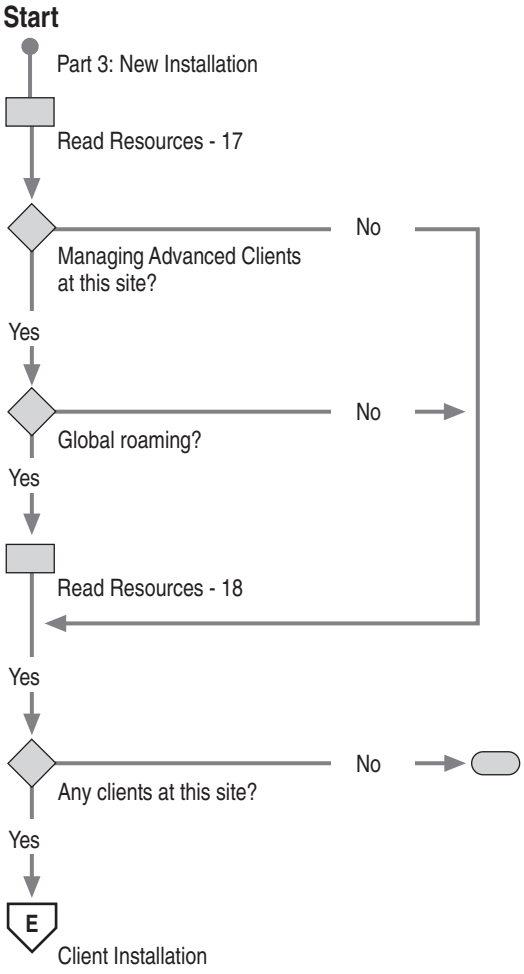
New Installation

After completing Parts 1 and 2, you might determine that you are deploying SMS 2003 for the first time, or that you do not have an existing SMS 2.0 site or SMS 2.0 clients that you wish to upgrade or migrate. In this case, you are deploying SMS 2003 as a new installation, and are following the deployment plan you developed in Parts 1 and 2.

Central Site Installation

As with any new installation of SMS 2003, the very first site that you deploy is a primary site. In this scenario, the first site is the central site. The flowchart in Figure 1.8 lists the steps for installing a central site.

Figure 1.8 Central site installation



It is recommended that you install a server locator point and a reporting point site system at the central site because site database information propagates from child sites to the central site. In large organizations, central sites typically do not manage SMS clients. If the site does manage SMS clients, then you need to set the boundaries appropriately. If you are managing Advanced Clients at the central site, and you intend to use global roaming throughout the SMS hierarchy, for example, you need to extend the Active Directory schema for SMS when you install the central site. After you have extended the Active Directory schema for SMS, it is extended for use by all SMS sites in the hierarchy in that Active Directory forest.

Note

There are other reasons for extending the Active Directory schema. For example, you might extend the Active Directory schema to take advantage of trusted root key exchange. The resources referenced in Resources 18 describe the reasons for extending the Active Directory schema.

Resources 17

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For a step by step description of the installation of an SMS site:	
Chapter 15	**Entire chapter recommended**

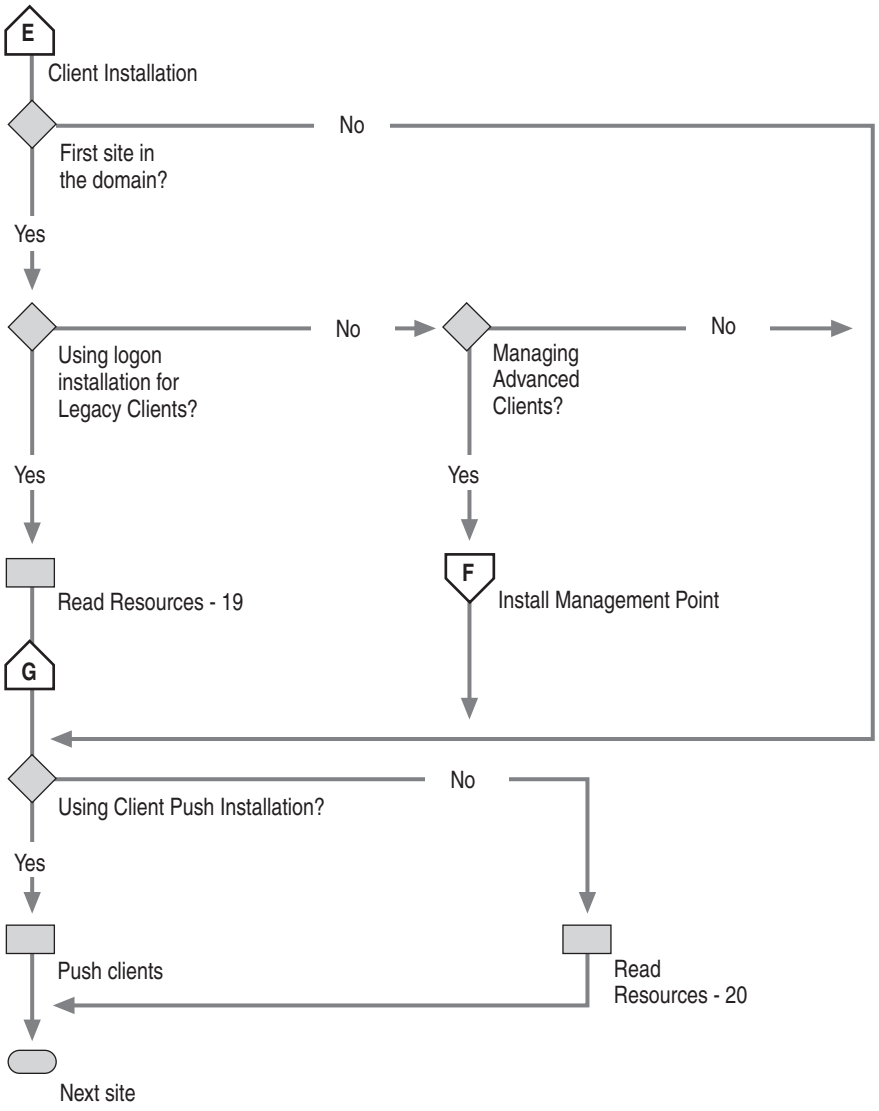
Resources 18

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For more information about extending the Active Directory schema:	
Chapter 10	**Extending the Active Directory Schema for SMS**
Chapter 15	**Extending the Active Directory Schema**

Client Installation

The flowchart in Figure 1.9 lists the steps and questions to consider when you install the SMS Legacy and Advanced Clients.

Figure 1.9 Client installation



If you are installing the Legacy Client using Logon Script-initiated Client Installation, the user logon scripts need to include Capinst.exe and identify the location of the client installation files. If you are installing the Advanced Client using Logon Script-initiated Client Installation, you need to install a management point to support those clients and modify the logon script accordingly. At this point, the flowchart in Figure 1.9 directs you to those specific steps (shown in Figure 1.10). After completing those steps, you return to this flowchart.

Note

If you are planning to install the Advanced Client software on computers using any installation method, you need to install a management point to support those computers as SMS clients.

If you are using the Client Push Installation method for either the Legacy or Advanced Client, you need to implement the correct accounts for the appropriate client types. For example, the Legacy Client requires a Client Connection Account and a Client Push Account. The Advanced Client requires an Advanced Client Network Access account and a Client Push account. There are two methods of pushing SMS client software to a computer — Client Push Installation and the Client Push Installation Wizard. Client Push Installation is started after you have configured and enabled it, and then when computers that require installation with Client Push Installation are discovered. Client Push Installation can also be started from a collection or resource by using the Client Push Installation Wizard. Table 1.4 describes the differences between Client Push Installation and the Client Push Installation Wizard.

Table 1.4 Client Push Installation Methods

Client Push Installation	Client Push Installation Wizard
Pushes client types: Legacy Client, Advanced Client, or Platform dependent. The option selected defines the site default.	Pushes Legacy Client, Advanced Client, or Platform dependent.
Ensures that all discovered computers within the site boundaries are installed with the SMS client.	Allows the installation of the SMS client on any computer that is found in the SMS Administrator console (for advanced clients, irrespective of whether they are within the site's roaming boundaries).
Does not push the client software again to existing SMS clients.	Supports pushing the client software again to existing clients for changes to site assignment and client component updates.
When enabled, runs until disabled by the SMS administrator.	Requires the SMS administrator to run the wizard.

Resources 19

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For more information about how to configure logon scripts:	
Chapter 17	**Logon Script-initiated Client Installation**

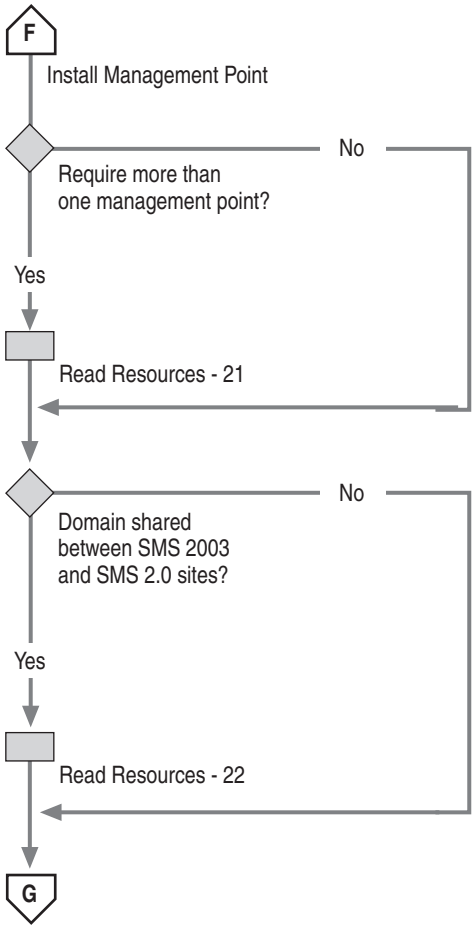
Resources 20

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For more information about other methods of deploying SMS clients:	
Chapter 17	**Installing and Configuring SMS Clients**

Management Point Installation

If you are supporting Advanced Clients in your SMS site, you need to install a management point in that SMS site. The flowchart in Figure 1.10 lists additional questions for you to consider when installing management points.

Figure 1.10 Management point installation



There is only one default management point for each SMS site. If you need to support multiple management points, you need to set up Windows Network Load Balancing between the management points. You might also choose to enable Microsoft SQL Server™ database replication between the SMS site database and the management point to reduce the load on the SMS site's computer that is running SQL Server, and facilitate faster response from management point servers.

You can configure the SMS 2003 site to use the Logon Script-initiated Client Installation method, and configure the SMS 2.0 site to run Capinst.exe from the SMS 2003 site. The logon scripts for the domain can contain a Capinst.exe command to install a Legacy Client or an Advanced Client. Use Capinst.exe with the **/AutoDetect**=<*script*> switch to determine which client type to install. For example, if the script you reference returns a value of 1, Capinst.exe installs the Advanced Client.

Resources 21

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For more information about how to configure management points and how to use NLB to support multiple management points:	
Chapter 8	**Management Point for Advanced Clients**

Resources 22

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about the command line options available to you when configuring a logon script-initiated installation:	
Chapter 17	**Logon Script-initiated Client Installation**

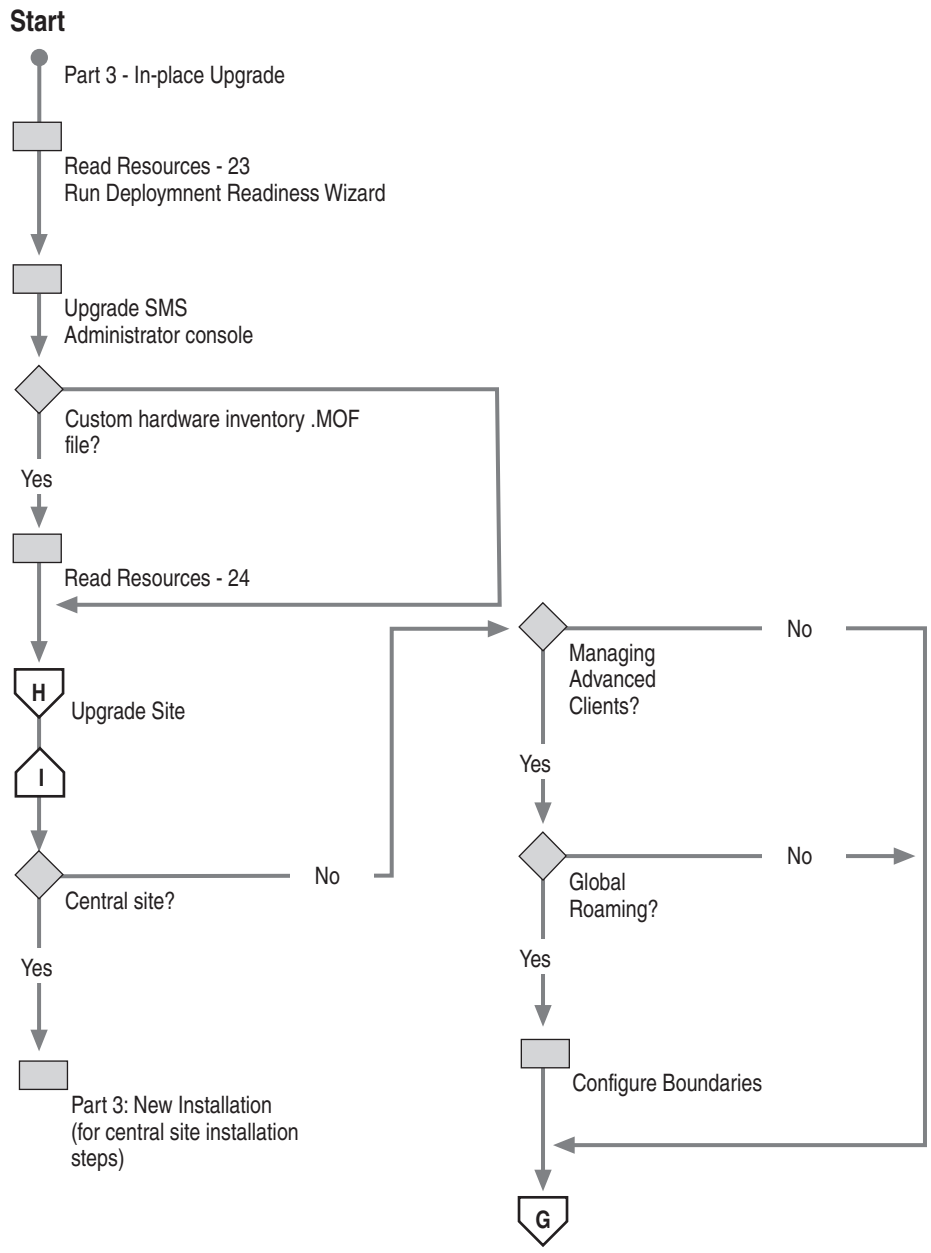
In-Place Upgrade

After completing Parts 1 and 2, you might determine that you can upgrade an existing SMS 2.0 site directly to SMS 2003 — an in-place upgrade. This section describes the in-place upgrade method of deploying SMS 2003. When you deploy SMS 2003 using the in-place upgrade method, the SMS site server and its site systems do not change their roles. An SMS site server that is assigned the CAP role remains a CAP after the upgrade has been completed. Also, SMS clients do not change their site assignments.

In-Place Upgrade Deployment Steps

The flowchart in Figure 1.11 lists the steps required to deploy SMS 2003 using an in-place upgrade.

Figure 1.11 In-place upgrade



You need to run the Deployment Readiness Wizard for every site that you intend to upgrade from SMS 2.0 to SMS 2003. The Deployment Readiness Wizard helps you determine what needs to be done to prepare your SMS 2.0 site for an upgrade. If the wizard finds errors, you must correct them and then run the wizard again before the upgrade can continue. After you correct all identified problems, you can upgrade the SMS site.

Customizations that you make to the SMS 2.0 SMS_def.mof file for hardware inventory are not migrated when you upgrade to SMS 2003. You must manually include those customizations in the SMS 2003 SMS_def.mof file that is created during the upgrade process. If you want to preserve the customizations you made to the SMS 2.0 MOF file, you need to save the existing file, and then merge it with the new file generated after the upgrade is complete.

If you plan to maintain a mixed-version hierarchy, consider using a standard SMS_def.mof throughout your hierarchy. Differences between the SMS_def.mof files at different sites in the hierarchy can lead to conflicting hardware inventory data. To prevent conflicts, ensure that each site in the hierarchy uses the same hardware inventory definitions.

Resources 23

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For detailed information about running the Deployment Readiness Wizard, and other considerations when planning to upgrade an SMS site from SMS 2.0 to SMS 2003:	
Chapter 11	**Resolve Issues Found by the Deployment Readiness Wizard**
Chapter 14	**SMS 2003 Deployment Readiness Wizard**

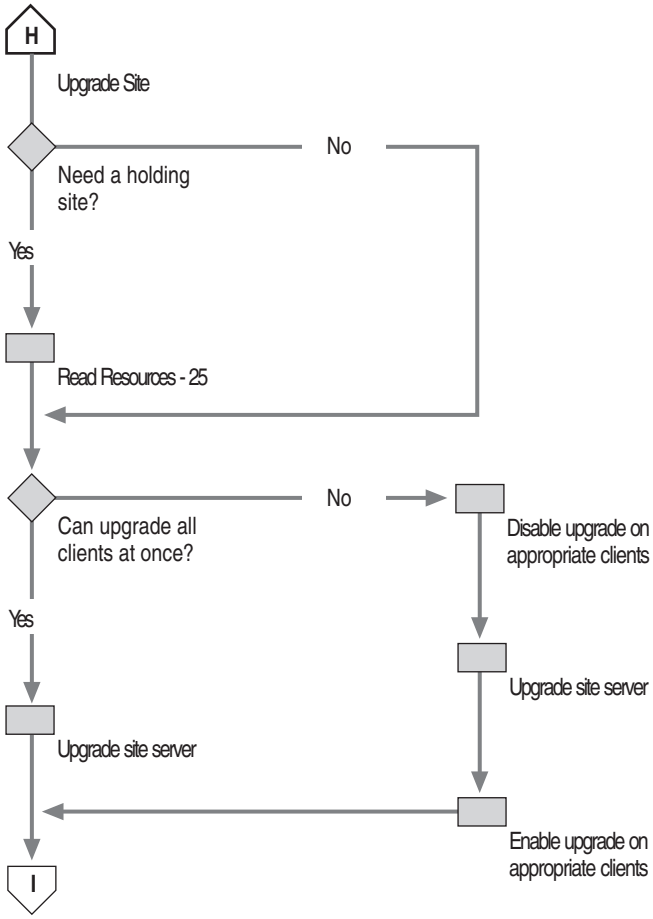
Resources 24

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For more information about how to standardize the SMS_def.mof files in your hierarchy:	
Chapter 6	**Hardware Inventory**
Microsoft Systems Management Server 2003 Operations Guide	
For more information about how SMS_def.mof is preserved during upgrades:	
Chapter 2	**Upgrading SMS and SMS_def.mof**

Upgrade Site

The next step shown in the flowchart in Figure 1.11 is to upgrade the site. The flowchart in Figure 1.12 lists the steps required to complete this part of the upgrade process.

Figure 1.12 Upgrade site



When you upgrade an SMS site from SMS 2.0 to SMS 2003, Class A and Class B clients assigned to that site automatically migrate to SMS 2003 Legacy Client. If you are upgrading from an SMS 2.0 site, you might have clients that fall into Class C as defined earlier in this chapter. Class C clients are not supported by SMS 2003, and they will become orphaned after the upgrade is complete.

The DRW will generate a warning message if it finds that the SMS 2.0 client is installed on any computers in the SMS site that run Windows 2000 or later operating systems. When you upgrade the SMS 2.0 site to SMS 2003, the Legacy Client is installed on those computers. This client is supported on Windows 2000 and later platforms primarily to assist with your migration of these clients to the Advanced Client rather than as a long-term enterprise solution. It is strongly recommended that you install the Advanced Client as soon as possible after the upgrade is complete so as to take advantage of the enhanced security and other benefits provided by the Advanced Client on these platforms.

In fact, the SMS 2003 status message system is designed to periodically notify you that such client configurations — Legacy Clients installed on computers running Windows 2000 or later — exist within your SMS site and should be upgraded to the Advanced Client. In addition, you can run the report or query named Computers Recommended for Advanced Client Upgrade that displays a list of these computers. You can use the query to create a collection to which you can advertise the Advanced Client installation to facilitate upgrading all your Legacy Clients to the preferred Advanced Client.

Class C clients require a holding site until they can be upgraded to a level supported by SMS 2003, or until you decide that you do not need to manage them. The holding site must be configured before you upgrade to SMS 2003. The Class C clients must be configured so that they do not attempt to migrate automatically to SMS 2003 clients. These are the basic steps to configure a holding site:

1. Deploy or choose an SMS 2.0 site that is a child of SMS site containing Class C clients. If Class C clients exist throughout the SMS hierarchy, you might make the holding site a child site of the central site.
2. Overlap the boundaries between the SMS site that you are upgrading and the holding site.
3. Allow the SMS clients to become assigned to both sites. Check the members of collections for both sites. When the members of collections for both sites are the same, this step is completed.
4. Wait until replication is complete between the holding site and its parent. Check the members of collections for both sites. When the members of collections for both sites are the same, this step is completed.
5. Upgrade the parent site to SMS 2003.
6. If the parent site is a central site, install a server locator point in the upgraded SMS site.

If your organization manages large numbers of Class A, B, and C clients, you might not be able to migrate all your clients at one time. In this case, use software distribution to run the Client Upgrade tool to disable migration on those clients that you are not ready to upgrade. When you are ready to upgrade those clients, you can use software distribution to run the Client Upgrade tool again to enable migration.

Resources 25

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For a detailed discussion about holding sites and other site upgrade considerations:	
Chapter 11	**Upgrade Strategies**
For a detailed discussion about the steps for upgrading an SMS site:	
Chapter 14	**Upgrading Primary Site Servers** **Upgrading Secondary Site Servers** **Performing Post-Upgrade Tasks**

At this point in the upgrade process, you return to the flowchart shown in Figure 1.11. The next question to consider is whether the site you are upgrading is a central site. If so, you can return to the flowchart shown in Figure 1.8. If not, you still need to consider whether you want to manage Advanced Clients at the site and whether you want to use global roaming as discussed in the "Client Installation" section earlier in this chapter, and then configure the roaming boundaries appropriately. Then you can proceed to install the Advanced Client software, following the steps and considerations listed in the flowchart shown in Figure 1.9.

Side-by-Side Upgrade

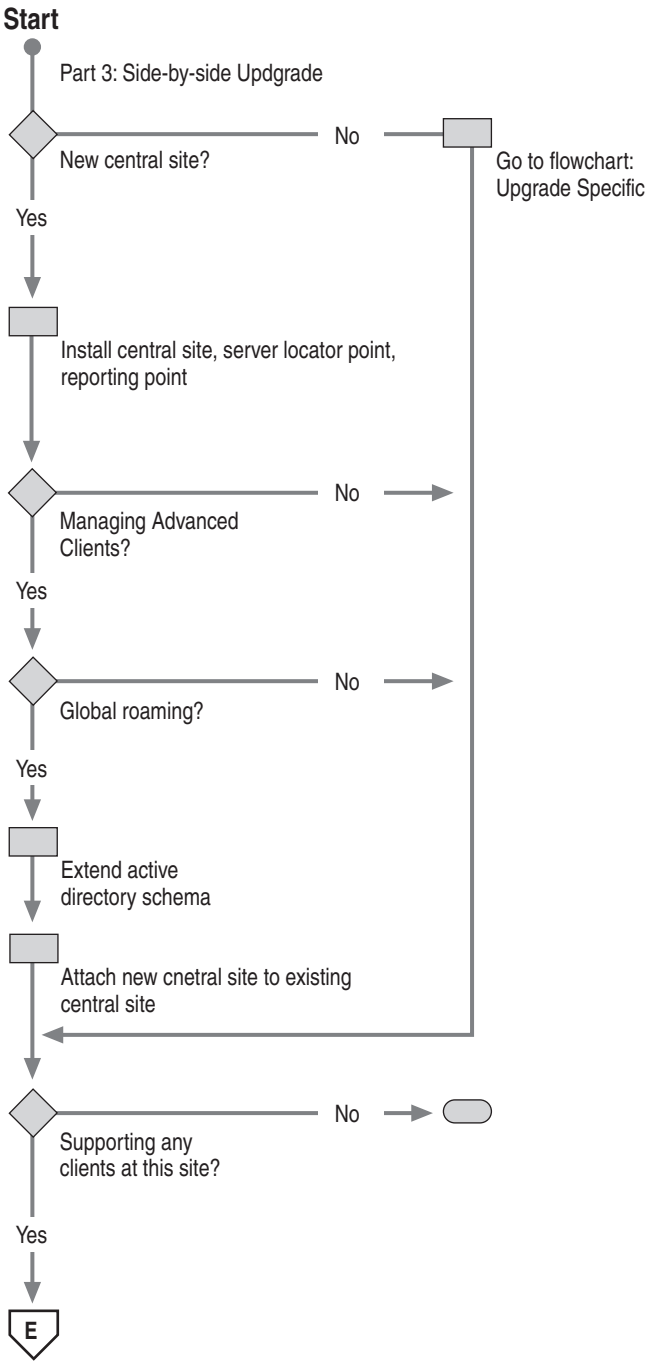
After completing Parts 1 and 2, you might determine that an in-place upgrade might not be the appropriate deployment method. You might intend to consolidate some or all of your existing SMS 2.0 sites, to change the structure of your existing SMS hierarchy, or to upgrade some or all of your server hardware. In this scenario, you can choose to deploy SMS 2003 using the side-by-side upgrade method. This section describes the side-by-side upgrade method of deploying SMS 2003. When you deploy SMS 2003 using the side-by-side upgrade method, you begin with the central site. You can either upgrade the existing SMS 2.0 central site to SMS 2003, or you can keep the existing central site and make it a child of a new SMS 2003 central site. In either case, you should implement an SMS 2003 site to act as a transition site for migrating existing SMS 2.0 clients that are Class A clients to the SMS 2003 Advanced Client.

Resources 26

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For more information about transition sites and other site upgrade considerations:	
Chapter 11	**Side-By-Side Hierarchy Upgrades**

The flowchart in Figure 1.13 lists the steps required to deploy SMS 2003 using a side-by-side upgrade.

Figure 1.13 Side-by-side installation



If you are upgrading the existing SMS 2.0 central site to SMS 2003, you follow the same basic steps that you would follow if you were upgrading the central site using an in-place upgrade. Those upgrade steps are listed in the flowchart shown in Figure 1.12.

If you are implementing a new central site, the process is similar to the one you follow for installing a new central site shown in the flowchart in Figure 1.8. However, after you have created the new SMS 2003 central site, you make the existing SMS 2.0 central site a child of the SMS 2003 central site. Then you can proceed to consolidate or upgrade your existing sites, install new SMS clients, and migrate existing SMS clients to the new SMS hierarchy as you designed it in Parts 1 and 2. Consolidate sites in the following manner:

- Make the site boundaries of the existing sites the roaming boundaries for the new site.
- Use software distribution to target Class A computers of the existing SMS hierarchy to install the Advanced Client software. You can use the predefined SMS package SMSClient.sms.
- Configure a holding site for any Class C clients that you must continue to manage

Post-Installation Considerations

After you upgrade a site, you must perform several additional tasks. You perform most of them from the SMS Administrator console. These tasks include:

Status filter rules after upgrading the site server to Windows Server 2003
If you have configured status filter rules to send a network message when an event occurs, and you upgrade the site server to Windows Server 2003, the status filter rules will no longer run. By default, the messenger service in Windows Server 2003 is disabled. To allow these status filter rules to run, enable and start the Messenger service.

Database maintenance and consistency checks It is a good idea to back up your upgraded site and to perform database consistency checks. For more information, see Chapter 13, "Maintaining and Monitoring SMS Systems."

This is a good time to schedule the backup task. For more information about backup and recovery, see Chapter 15, "Backup and Recovery."

Differences between the SMS_def.mof files at different sites of the same version in the hierarchy can lead to conflicting hardware inventory data. To prevent conflicts, you should make sure that each site of the same version in the hierarchy uses the same hardware inventory definitions.

For more information about how to standardize the SMS_def.mof files in your hierarchy, see Chapter 6, "Understanding Interoperability with SMS 2.0," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

For more information about how to restore your customized SMS_def.mof files after you upgrade, see Chapter 2, "Collecting Hardware and Software Inventory."

Site configuration You must configure the site settings for all new SMS 2003 sites. This applies to newly installed SMS 2003 sites and to sites upgraded to SMS 2003 from SMS 2.0. Configuration settings from SMS 2.0 are preserved during an upgrade. For example, you must configure the site boundaries and enable client installation methods to upgrade clients and populate the SMS site database.

In general, perform post-upgrade tasks in the following order:

1. Configure all site settings.
   - Assign new site system roles.
   - Specify the IP subnets or Active Directory sites that define your site boundaries.
   - Enable resource discovery methods.
2. Enable client installation methods.

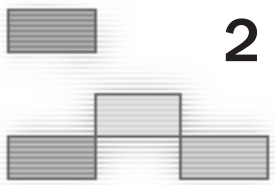
Finally, after planning the strategy for upgrading your SMS hierarchy, you must plan for features you want to use in SMS 2003. You must determine if your SMS 2.0 clients use features that are not supported SMS 2003. You also must determine if there are any requirements you must meet for new SMS 2003 features.

Resources 27

Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide	
For more information post-upgrade planning for SMS features:	
Chapter 11	**Post-upgrade Migration Planning**

CHAPTER 2

Collecting Hardware and Software Inventory



By collecting hardware and software inventory data with Microsoft® Systems Management Server (SMS) 2003, you can build a rich database containing detailed information about the computers in your organization.

Overview

You can employ several SMS features to use the data that SMS collects by using hardware inventory and software inventory. For example:

- You can build queries that include computers based on their hardware configuration or installed software. The queries are useful to technical analysts and others who want to proactively prevent problems by checking for computers with configuration problems, such as insufficient disk space.
- You can build collections with queries that include computers based on their hardware configuration or installed software. Those collections can then be used to advertise software packages to computers that require the software and are capable of supporting it.
- You can produce reports that display useful hardware configuration or installed software details. The reports are useful to managers, systems analysts, and others who need to make decisions based on information about the current computer infrastructure.
- You can use the SMS Resource Explorer to view the complete inventory data for individual computers. This view of individual computers is especially useful when remotely troubleshooting computer problems.

SMS software inventory can also collect files, not just details about the files, from SMS client computers. With file collection, you specify a set of files to be copied from clients to the SMS site that the clients are assigned to.

Chapter 3, "Understanding SMS Features," of the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* introduces hardware and software inventory in more detail. That chapter also explains inventory resynchronization, delta inventory collection, and similar topics that are key to the successful use of the SMS inventory features.

This chapter prepares you to implement and use SMS inventory. In the future, you might have some special requirements when using the Resource Explorer, or you might want SMS to collect information about your computers that requires special extensions to the inventory collection processes. At that time, you should read Chapter 3, "Advanced Inventory Collection."

Distinguishing Between Hardware Inventory and Software Inventory

When working with SMS inventory features, remember the distinctions between hardware inventory and software inventory. The primary distinction between the two inventory mechanisms is how they work.

Software inventory works by scanning the disks on each computer to find files and gather information about files. You can also configure software inventory to collect specific files when it finds them.

Hardware inventory works by querying Windows Management Instrumentation (WMI) for all data from certain WMI classes. For more information about WMI, see Appendix B, "Windows Management Instrumentation." WMI includes classes for operating system configuration and entities (such as user accounts), installed software, software configuration, and other objects (such as for the logged on user). These classes are supplements to hardware classes. Hardware inventory collects information about many things besides hardware. For example, it can inventory software by collecting details about programs listed in **Add or Remove Programs** in Control Panel or programs that have been installed using Windows Installer.

Because hardware inventory collects a wide variety of data, you might determine that most of your inventory needs can be served by hardware inventory collection alone. Also, with hardware inventory, you can customize inventory to collect more data or different data, as described in Chapter 3, "Advanced Inventory Collection." Software inventory is useful when you require information about the files on the disks, not necessarily about the software that has been installed. In that sense, software inventory could be called "file inventory."

Examples of commonly used inventory classes and the inventory methods that must be enabled to collect them are included in the "Reviewing the Inventory Data" section later in this chapter.

In This Chapter

- Hardware Inventory Administrative Tasks
- Software Inventory Administrative Tasks
- Using Resource Explorer to View Inventory Data
- Other Considerations for Collecting Inventory

Hardware Inventory Administrative Tasks

There are several tasks you can do to manage hardware inventory, including:

- Enabling and disabling hardware inventory.
- Scheduling hardware inventory.
- Configuring hardware inventory rules.

The "Viewing Hardware Inventory" section later in this chapter describes how to view collected inventory data by using Resource Explorer.

Note

Hardware inventory can use considerable network capacity. The network capacity required to run hardware inventory depends on the number of SMS clients you have, how frequently you schedule hardware inventory, and the size of the inventory data you collect. If you expect hardware inventory to slow network activity significantly, consider running this process during nonpeak hours.

Enabling and Disabling Hardware Inventory

Hardware inventory is always installed on the SMS site server. You can enable or disable the hardware inventory client agent any time by using the SMS Administrator console. The hardware inventory client agent is always installed on Advanced Clients. It is installed on Legacy Clients only when the client agent is enabled. In the SMS hierarchy, inventory data is forwarded from child sites to parent sites to allow for centralized administration. If child sites have hardware inventory enabled, their hardware inventory data is propagated to the parent site even if the parent site has hardware inventory disabled.

To enable or disable hardware inventory, navigate to the **Hardware Inventory Client Agent** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ <site code - site name>
    └ Site Settings
     └ Client Agents
```

In the details pane, right-click **Hardware Inventory Client Agent** and click **Properties**. To enable hardware inventory, select **Enable hardware inventory on clients**. To disable hardware inventory, clear **Enable hardware inventory on clients**. Then, set the schedule for hardware inventory and the maximum custom Management Information Format (MIF) file size. MIF files are used by SMS to extend SMS inventory collection and to provide detailed software distribution status. For more information about using MIF files to collect supplemental inventory information, see Chapter 3, "Advanced Inventory Collection."

When the hardware inventory agent is installed and enabled on Legacy Clients, hardware inventory is collected after 10 minutes and then according to the hardware inventory schedule that you specify in the agent. When the hardware inventory agent is enabled on Advanced Clients, hardware inventory only runs according to the hardware inventory schedule you specify.

Scheduling Hardware Inventory

By default, hardware inventory runs once every seven days. You change the hardware inventory schedule by setting the time of day or frequency that best suits your requirements. You can change hardware inventory settings at any time. The next inventory cycle after the client picks up the new settings for the site reflects your changes.

You schedule the hardware inventory process by configuring settings in Hardware Inventory Client Agent properties. Begin by navigating to the **Hardware Inventory Client Agent Properties** dialog box as directed in the "Enabling and Disabling Hardware Inventory" section earlier in this chapter, and select the best schedule for your SMS site. To schedule hardware inventory, you can either select an interval, or you can specify a start date and time and a recurring schedule. For more information about scheduling hardware inventory, see the SMS Help.

Important

If an Advanced Client roams to a secondary site and connects to a proxy management point, its inventory is propagated to the primary parent site of the secondary site. If the SMS addresses at the secondary site are configured to forward the inventory data to the parent site after the roaming Advanced Client has returned to its assigned site and reported inventory directly, an inventory resynchronization can be caused for the client. If many clients do this, significant network and server activity could result. To avoid this problem, set the inventory schedule to be less frequent than site-to-site communications.

Forcing Hardware Inventory on an SMS client

To run hardware inventory immediately on a single client, use the **Systems Management** icon in Control Panel on the client computer.

To force hardware inventory on the Advanced Client

1. In Control Panel, double-click the **Systems Management** icon.
2. On the **Actions** tab, click **Hardware Inventory Cycle**.
3. Click **Initiate Action**.

To force hardware inventory on the Legacy Client

1. In Control Panel, double-click the **Systems Management** icon.
2. On the **Components** tab, click **Hardware Inventory Agent**.
3. Click **Start Component**.

Forcing hardware inventory does not disrupt the normal hardware inventory cycle if it is set to run on a full schedule (at a specific time and day, for example). In that case, the regularly scheduled hardware inventory still runs at the time scheduled in the hardware inventory agent. However, if inventory is set to run on a simple schedule of once per day, for example, then the next inventory cycle is run 24 hours from the time the inventory is forced, and every 24 hours thereafter.

Enabling and Disabling MIF Collection

You can use IDMIF and NOIDMIF files to collect supplemental information about SMS client computers or other resources during hardware inventory, as described in Chapter 3, "Advanced Inventory Collection."

Collecting IDMIFs or NOIDMIFs can be a security risk, so you can disable their collection if that risk is significant to you. For more information about IDMIF and NOIDMIF security issues, see the "Inventory Collection" section in Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Newly installed SMS 2003 sites have MIF collection disabled by default. SMS 2003 sites that have been upgraded from SMS 2.0 have MIF collection enabled by default. Disabling hardware inventory MIF collection does not disable software distribution status MIF collection. For more information about software distribution status MIFs, see Chapter 5, "Distributing Software."

To enable or disable MIF collection

1. Click the **MIF Collection** tab in the **Hardware Inventory Client Agent Properties** dialog box.
2. Select or clear the options to collect IDMIF or NOIDMIF files for the Legacy Client and Advanced Client.

Caution

When NOIDMIF collection is disabled, the data collected using NOIDMIFs is deleted from the SMS site that the clients are assigned to.

Configuring Hardware Inventory Rules

By default, SMS hardware inventory collects a rich set of information about your client computers by using WMI. WMI can also provide more information. Hardware inventory is configured to collect the data that is most likely to be useful to you. You can review the hardware inventory configuration to ensure that SMS is collecting the data that you require. You can adjust the SMS hardware inventory configuration to collect more or less data accordingly.

The SMS hardware inventory configuration is adjusted by manipulating a file named SMS_def.mof. The SMS_def.mof file is stored in the \SMS\Inboxes\Clifiles.src\Hinv folder on the SMS site server. The following two sections provide information about how to modify this file. You can also extend SMS hardware inventory by defining additional classes for WMI to collect, and by adding new classes to the SMS_def.mof file. For more information, see Chapter 3, "Advanced Inventory Collection."

Your changes to SMS_def.mof are automatically propagated to all clients at the SMS site. They are not propagated to any other sites. You must make the same changes to the SMS_def.mof at other sites, or copy the SMS_def.mof to those sites. Be careful when copying the SMS_def.mof from one site to a site that might be running a different version or service pack of SMS. The version of the SMS_def.mof that you copy might not include changes you or Microsoft have made in the SMS_def.mof at the destination site.

Important

If you modify the SMS_def.mof file or create custom MIF files (as described in Chapter 3, "Advanced Inventory Collection") to add information to inventory, consider the performance effects. Adding certain information (for example, adding the Win32_LogEvent, Win32_Account, or Win32_Directory classes) can slow network and system performance appreciably.

Advanced Clients download new hardware inventory rules when Advanced Client policy is refreshed. By default, this is once per hour. Legacy Clients download new hardware inventory rules when their client refresh cycle is run. By default, this is once every 25 hours. When the clients have the new hardware inventory rules, the next hardware inventory is collected according to the modified SMS_def.mof file, as long as it is syntactically correct. Otherwise, the previous version of SMS_def.mof is used.

Do not place custom SMS_def.mof files on Legacy Clients or CAPs. If you do, those files are used temporarily and then overwritten. At each daily client refresh cycle, the SMS_def.mof on the SMS site server is compared with the copy on the client, and if these copies are different, the copy on the server is replicated to the client, overwriting any custom SMS_def.mof file that exists on the client. Copies of the SMS_def.mof file also exist on Legacy Clients, but you should not modify them. The SMS client automatically updates these copies when necessary.

If you make changes to the SMS_def.mof, you must back up the file before upgrading the site to a newer version of SMS. If Microsoft has not made any changes to the SMS_def.mof in the new version of SMS, you can restore your SMS_def.mof. You can determine whether Microsoft has made any changes to the SMS_def.mof by comparing it to the original SMS_def.mof of the previous version of SMS. If Microsoft has made changes to the SMS_def.mof, you must apply your changes to the new version of the SMS_def.mof.

For example, when a service pack is available for SMS 2003, you should compare its SMS_def.mof with the SMS_def.mof that was originally installed with SMS 2003. If there are no differences, you can restore your SMS_def.mof in place of the one that is included in the service pack. Otherwise, you should apply your changes to the version in the service pack.

Keep a backup copy of the SMS_def.mof file. You can configure the Backup SMS Site Server procedure in the SMS Administrator console. The SMS_def.mof file is backed up as part of this task. Or, you can back up the SMS_def.mof file separately, ideally whenever you change the SMS_def.mof file. For more information about how SMS_def.mof is preserved during upgrades, see the "Distributing SMS_def.mof" section later in this chapter. For more information about using the backup task, see Chapter 15, "Backup and Recovery."

Note

The Advanced Client does not use a copy of SMS_def.mof on the client. However, SMS_def.mof is stored in the SMS site database as soon as changes are made, and then converted into Advanced Client policy. Editing SMS_def.mof is the means for configuring hardware inventory for all clients in SMS, although you do not find SMS_def.mof on Advanced Clients.

Editing SMS_def.mof

To edit SMS_def.mof file, use a text file editor to change the class and property reporting settings.

Each property and class has an SMS_Report flag. To include a property or class in inventory, set the SMS_Report flag to TRUE. To remove a property or class from inventory, set the SMS_Report flag to FALSE.

SMS_def.mof starts with the definition of namespaces, base classes, and providers that are needed by the Hardware Inventory Agent and WMI. The rest of the file defines the classes that the Hardware Inventory Agent can collect data about. The syntax of the SMS_def.mof is the same as any other MOF file. However, it also includes class and property qualifiers that are used by the Hardware Inventory Agent.

Note

Group names can use double-byte character set names. If this is done, the SMS_def.mof file must be saved as a Unicode file.

Class Qualifiers:

- SMS_Report is an optional Boolean value indicating whether or not the class is to be collected by SMS inventory. Its default value is FALSE.
- SMS_Group_Name is an optional name of the property group to be used when collecting the class. By default, it is the WMI class name as it appears in SMS_def.mof.
- SMS_Class_ID is a required SMS class identifier string associated with the property group. The class identifier is a three-part string delimited by vertical bars. The first part is the vendor, the second part is a group name, and the third part is a version number.
- SMS_Namespace is an optional Boolean value indicating whether the provider for this class is located in the root\CIMv2\SMS namespace. This must be set to TRUE for any class whose data is provided directly to the SMS reporting class. If SMS_Namespace is set to FALSE, or not specified, the data is collected from the root\CIMV2 namespace or the namespace specified in using the Namespace class qualifier.
- Namespace is an optional value indicating where the hardware inventory agent should look for the data class. Namespace only applies to Advanced Clients. Legacy Clients ignore this class qualifier.

Property Qualifiers:

- SMS_Report is an optional Boolean value (TRUE, FALSE) indicating whether or not the property is to be included in SMS inventory. The default is FALSE. For key properties, this qualifier is ignored on Legacy Clients. Keys are always reported on Legacy Clients.
- SMS_Units is an optional string that informs the Hardware Inventory Agent to perform a conversion between data provided by WMI into a form SMS can use. If the data is in a normal property, the property is rejected. If the data is in a key property, the instance is rejected.

  For example, SMS cannot use 64-bit integers, so in the case of disk size, the qualifier "SMS_Units("Megabytes")" is used. The agent translates the WMI value in bytes into the appropriate representation in MB. This qualifier is ignored for non-integer properties.

  Another example is using the DateString value for the SMS_Units qualifier for WMI date-time intervals. These are in the format ddddddddHHMMSS.mmmmmm:000. SMS requires the DateString qualifier to convert and use WMI time-intervals.

  Possible SMS_Units values:

  - KB — divides by 1024
  - MB — divides by (1024 × 1024)
  - HexString — converts number to hex strings. For example, decimal value 161 is converted to string "0×A1."
  - DecimalString — SMS cannot use 64-bit integers, so this converts WMI uint64 values to string values

- Seconds — divides time values in milliseconds by 1000
- DateString — converts time interval strings. For example, a DateTime value of "00000008061924.000000:000" turns into the string "8 Days 08:15:55 Hours".

For information about the specific classes and properties in the SMS_def.mof file, see the SMS SDK. The SMS SDK is available as part of the Platform SDK, which is available from Microsoft Developer Network (MSDN), or at http://www.microsoft.com/smserver.

Distributing SMS_def.mof

Whenever the SMS_def.mof file is changed on a primary site server (including when SMS is upgraded, if the SMS_def.mof has changed in the newer version of SMS), SMS loads its contents into the SMS database so that Advanced Clients can request them as policy from the management point. The SMS_def.mof is also downloaded to CAPs so that Legacy Clients can acquire it. This is also done at secondary sites. Both clients download the changes during their daily client refresh cycles.

While SMS_def.mof is loaded into the SMS site database, SMS backs up the SMS_def.mof to the \SMS\data\hinvarchive folder. If the SMS_def.mof is valid, it is backed up as SMS_def.mof.bak. If an SMS_def.mof.bak already exists, SMS_def.mof.bak is first backed up as SMS_Def.mof.bk0. If an SMS_def.mof.bk0 already exists, it is first backed up as SMS_def.mof.bk1. This continues to SMS_def.mof.bk4.

If the SMS_def.mof is not valid, it is backed up as SMS_def.mof.bad.bak. If an SMS_def.mof.bad.bak already exists, SMS_def.mof.bad.bak is first backed up as SMS_Def.mof.bad.bk0. If an SMS_def.mof.bad.bk0 already exists, it is backed up as SMS_def.mof.bad.bk1. This continues to SMS_def.mof.bad.bk4.

Upgrading SMS and SMS_def.mof

If you have upgraded from SMS 2.0 to SMS 2003, you can compare the SMS_def.mof in SMS\Inboxes\Clifiles.src\Hinv to SMS_def.mof.bak or SMS_def.mof.bk0 in \SMS\data\hinvarchive to see if you have made any customizations that you want to reapply to the SMS 2003 SMS_def.mof. Do not copy SMS_def.mof.bak over SMS_def.mof. You lose the Microsoft changes to SMS_def.mof that are introduced with SMS 2003.

Note

If you are upgrading to SMS 2003, carefully compare the SMS 2003 SMS_def.mof to your previous SMS_def.mof. Numerous changes have been made to the SMS 2003 SMS_def.mof to include additional useful classes, to reflect changes in WMI, and to remove less useful classes.

When a Legacy Client receives new hardware inventory rules, it generates a complete hardware inventory instead of a delta inventory of changes only. The SMS site server deletes data for the client for any classes not included in the complete inventory from the client (which also means that the classes were not included in the new SMS_def.mof). The history data for any such classes is not deleted. If you had made customizations to hardware inventory, the data for those customizations is lost when you upgrade to SMS 2003 (and its new SMS_def.mof) until you reimplement those customizations and allow time for the clients to run the next hardware inventory cycle. The Advanced Client does not generate a full inventory when it receives new hardware inventory rules. It always generates a delta inventory.

Note

The SMS 2003 SMS_def.mof includes some classes that you might have added as hardware inventory extensions (for example, a list of the installed programs in the **Add or Remove Programs** icon in Control Panel). If you have made hardware inventory extensions in SMS 2.0, you should review the SMS 2003 SMS_def.mof to see if it includes your extensions. If it does, you do not need to re-implement your extensions.

You can avoid losing the data from your hardware inventory customizations (and one of the two full inventory cycles) by disabling the hardware inventory client agent before beginning the SMS site upgrade. When the upgrade is completed, reimplement your customizations in the SMS 2003 SMS_def.mof, and then enable the Hardware Inventory Client Agent. SMS clients still generate one full hardware inventory because of the Microsoft changes to SMS_def.mof, but the data for your customizations is not temporarily lost, and a second full hardware inventory is not required.

Important

If you implemented your SMS 2.0 hardware inventory extensions without changing the SMS_def.mof, be sure to adjust those extensions so that the reporting classes are included in the SMS_def.mof. The data class definition and population can still be included in your customization. For more information, see Chapter 3, “Advanced Inventory Collection.”

Software Inventory Administrative Tasks

This section describes the tasks you can do to manage the software inventory process:

- Enabling and disabling software inventory
- Scheduling software inventory
- Configuring software inventory rules
- Configuring file collection

- Managing inventory names
- Controlling software inventory on servers

The "Viewing Software Inventory" section later in this chapter describes how to view collected inventory data by using Resource Explorer.

Note

Software inventory can use considerable network capacity. The amount of network capacity used depends on the number of SMS clients you have, how frequently you schedule software inventory, and the size of the files you collect (if any). If you expect that software inventory will significantly affect network activity, consider running this process during nonpeak hours.

Enabling and Disabling Software Inventory

Software inventory is always installed on the SMS site server. You can enable or disable the software inventory client agent any time by using the SMS Administrator console. The software inventory client agent is always installed on Advanced Clients. It is installed on Legacy Clients only when the client agent is enabled. In the SMS hierarchy, inventory data is forwarded from child sites to parent sites to allow for centralized administration. If child sites have software inventory enabled, their software inventory data is propagated to the parent site even if the parent site has software inventory disabled.

To enable or disable software inventory, navigate to **Software Inventory Client Agent** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ <site code - site name>
    └ Site Settings
     └ Client Agents
```

In the details pane, right-click **Software Inventory Client Agent**, and then click **Properties**. To enable software inventory, select **Enable software inventory on clients**. To disable software inventory, clear **Enable software inventory on clients**.

When the software inventory agent is installed and enabled on Legacy Clients, software inventory is collected after 20 minutes and then according to the software inventory schedule. When the software inventory agent is enabled on Advanced Clients, it runs only according to the software inventory schedule.

Scheduling Software Inventory

By default, SMS software inventory runs once every seven days. You can change the software inventory schedule by setting the time of day and frequency that best suits your requirements. The software inventory agent does many disk reads on each SMS client to collect software inventory. In some cases, users might notice a slowdown on their computer as result of this activity. You should test software inventory in your test lab using typical user configurations to see if this might be an issue for your users. At large sites, software inventory collection can result in a significant amount of network activity. You can schedule software inventory to always occur when the client agent activity has the least impact on users.

Schedule software inventory by configuring settings in the **Software Inventory Client Agent Properties** dialog box. Navigate to the **Software Inventory Client Agent Properties** dialog box as directed in the "Enabling and Disabling Software Inventory" section earlier in this chapter, and specify the best schedule for your SMS site. There are two ways to schedule software inventory. You can either select an interval, or you can specify a start date and time, and a recurring schedule. For more detailed information about scheduling software inventory, see the SMS Help.

Forcing immediate software inventory on a client

To run software inventory immediately on a single client, use the **Systems Management** icon in Control Panel.

► To force a software inventory on the Advanced Client

1. In Control Panel, double-click the **Systems Management** icon.
2. On the **Actions** tab, click **Software Inventory Cycle**.
3. Click **Initiate Action**.

► To force a software inventory on the Legacy Client

1. In Control Panel, double-click **Systems Management**.
2. On the **Components** tab, click **Software Inventory Agent**.
3. Click **Start Component**.

Forcing software inventory does not disrupt the normal software inventory cycle. The regularly scheduled software inventory still runs at the time scheduled in the Software Inventory Agent.

Configuring Software Inventory Rules

By default, the Software Inventory Client Agent inventories all .exe files on all SMS client hard disks, but you can also specify other file types or folder trees for software inventory.

To configure software inventory rules

1. In the SMS Administrator console, click the **Inventory Collection** tab in the **Software Inventory Client Agent Properties** dialog box.
2. Click the **New** icon, and then type the name of a file you want to inventory.

   You can type exact file names (such as Autoexec.bat), or you can use wildcards. For example, you can inventory all files of a certain extension, such as *.zip. Any valid use of wildcards for the DIR command is valid in this dialog box.
3. By default, all hard disks on the SMS client are inventoried. If you want to inventory a folder or folder tree, click the **Set** button. In the **Path Properties** dialog box, click **Variable or path name**, and then specify a folder or folder tree.

   A variable is an environment variable, such as %*Windir*%. You can also specify whether subfolders should be searched by setting **Search subdirectories**. Wildcards can also be used in the last part of the path, for example, %*ProgramFiles*%\Microsoft Visual*.

   > **Important**
   >
   > The Software Inventory Agent supports both system and user environment variables, but the user environment variables are for the security context the agent runs in, not the context of the currently logged on user.
   >
   > Also, the value of the environment variable must not contain an environment variable. For example %*temp*% cannot be used if its value is "%*Windir*%\temp."
4. Set **Exclude encrypted and compressed files** if you do not need to inventory them. By default, this option is enabled. This setting is particularly important if you are collecting product details during software inventory. Product details are contained within the files, so encrypted and compressed files must be decrypted and decompressed, which can use considerable computer resources on the SMS clients. If the local system account (or a group that contains the local system account) is not given administrative rights to the encrypted files, SMS cannot decrypt them.
5. Repeat steps 2 through 4 for all the inventory rules you require. Additional rules impose additional workload on the clients and might create additional network traffic or workload on the SMS servers. You should carefully consider the need for each additional rule. There is a maximum limit of 64 rules.
6. Set the level of reporting details you want to collect using software inventory by setting **File details** and **Product details**.

   If you set **Product details**, the following properties are collected for each file:

   - Manufacturer name
   - Product name
   - Product version
   - Product language

If you set **File details**, the following properties are collected for each file:

- File name
- File path
- File size
- Modified date

If you set both **File details** and **Product details**, the following properties are also collected for each file:

- File description
- File version

Note

File details are obtained by scanning folder entries. Product details are obtained by opening the files. File details are more efficient because fewer disk reads are required. Also, because the files do not need to be loaded into memory to obtain the product details, they do not have to be scanned by antivirus software that might be running on the clients. However, because it is much harder to hide files by changing the product name than by changing the file name, collecting product details can provide more accurate results if your users might try to hide programs by renaming them.

You cannot clear both the **Product details** and **File details** options. At least one of these sets of details must be collected.

Configuring File Collection

File collection copies files from SMS clients to the SMS site server. You use software inventory to collect files from clients and store them at the primary site server that the clients are assigned to. The files are collected the next time software inventory runs after the file collection rule is created and propagated to clients. They are not collected again until inventory collection runs and the files have changed. You must specify the files you want to collect. When you do, you can use wildcard characters so that you collect all initialization files (*.ini), for example. You can also specify multiple variations of a file, such as Status*.doc.

To configure file collection

1. Select the **File Collection** tab in the **Software Inventory Agent Properties** dialog box.
2. Click the **New** icon, and then type the name of a file you want to collect.

   You can type exact file names, or you can use wildcards (such as *.zip). Any valid use of wildcards for the DIR command is valid in this dialog box. Wildcards can also be used in the last part of the path, for example, %*ProgramFiles*%\Microsoft Visual*.

> **Note**
> The value of the environment variable must not contain an environment variable. For example %*temp*% cannot be used if its value is "%*Windir*%\temp."

3. By default, all hard disks on the SMS clients are scanned for files to collect. If you want to scan a particular folder or folder tree, click the **Set** button. In the **Path Properties** dialog box, click **Variable or path name**, and then specify a folder or folder tree.

   A variable is an environment variable, such as %*Windir*%. By setting **Search subdirectories**, you can also specify whether subfolders should be searched.

4. Set **Exclude encrypted and compressed files** if the desired files are not encrypted or compressed. If the local system account (or a group that contains the local system account) is not given administrative rights to encrypted files, SMS cannot decrypt or collect them. Excluding these files also makes the collection process more efficient.

5. Set the **Maximum size (KB)** for the files to be collected. This is the maximum size of the file or files collected for this rule. If the total size of the files collected by this rule exceeds this value, none of the files are collected.

> **Note**
> When SMS sends a large volume of collected files across the network, network performance can suffer. To minimize this problem, you can use the **Maximum Size (KB)** option, restrict the path so that you collect only copies of files from the desired folder tree, or schedule software inventory when network traffic is lightest. The sum of the **Maximum Size (KB)** options is indicated as the **Maximum traffic per client (MB)** value on the **File Collection** tab.
>
> Also, during the collection process SMS makes a temporary copy of the files being collected. Sufficient disk space must be available for the copies. If multiple file collection rules apply to a file, and it is within the size limitation of one rule but not another, the file is not collected.
>
> Be aware that collecting all .dll files from each client can create considerable network traffic.

Managing Inventory Names

When software is developed, individual files are often identified with the product name and manufacturer name in a header. These properties are displayed when you view the properties of a file in Windows Explorer.

However, the product name and manufacturer name are sometimes misspelled or recorded inconsistently in headers. For example, “Microsoft,” “Microsoft Corporation,” and “Micorsoft” might all be found in different header blocks yet refer to software created by the same manufacturer — Microsoft Corporation. In SMS, inventory name conversion rules are used to map misspellings or inconsistencies in the inventoried software product or manufacturer names. You can use conversion rules to map the misspelled and inconsistent names to any name you choose.

For example, in SMS Resource Explorer, the manufacturer name is one of the nodes that software is grouped under, so if each variation of one manufacturer was left as is, there could be a lot of nodes for each manufacturer, even though they are essentially the same. The same is true when running queries or reports where software is grouped by manufacturer name. To avoid this, set inventory names.

To set inventory names

1. Click the **Inventory Names** tab in the **Software Inventory Agent** dialog box.
2. Select either **Product** or **Manufacturer** from the **Name type**.
3. Select the **Display name** if the product or manufacturer already has an entry. Otherwise, click the **New** icon above the **Display name** list, and then type the name of a product or manufacturer you want the names to be consolidated to.
4. Click the **New** icon above the **Inventoried names** list and then type the name of a product or manufacturer as it would be inventoried. Use “%”as a wildcard in the name where the name might vary by zero or more characters. Use “_” as a wildcard in the name where the name might vary by only a single character.

Controlling Software Inventory on Servers

Servers often have large disk drives with many files that are accessed by many users. Managing servers with SMS and even inventorying the installed software might be useful, so installing the SMS client on servers can be valuable. However, inventorying files on the shared disk drives can take considerable resources on the server and generate considerable network traffic and workload on the SMS servers.

To avoid the overhead of running software inventory on large disks, you can create a hidden file named Skpswi.dat and place it in the root folder of each disk drive that you want excluded from software inventory. Software inventory does not scan these drives unless the Skpswi.dat file is removed.

You can also place a Skpswi.dat file in the folder that is at the top of the path of a software inventory collection rule. For example, if you have a rule to inventory “\Program Files,” that entire folder tree is skipped on any SMS client that has a Skpswi.dat file in the “\Program Files” folder.

Note

Skpswi.dat also applies to file collection. Disks with a Skpswi.dat file are not scanned to find files that are to be collected.

SMS automatically excludes the Recycle Bin from inventory on all SMS clients.

You might find that software inventory scans folders that include secondary copies of files. This is especially true if you scan compressed folders, which includes the operating system DLL cache and service pack uninstall folders. If you do not want to inventory such folders, place a Skpswi.dat file in those folders on your SMS clients.

Using Resource Explorer to View Inventory Data

Resource Explorer is a tool in the SMS Administrator console that displays the collected inventory data. When you invoke Resource Explorer, it opens a window that displays the information collected by hardware inventory and software inventory.

If a resource is also an SMS client, and if you are collecting hardware inventory at your site, the records for that resource include a list of the hardware installed on the client and similar details. If you are collecting software inventory, the records also include the software listing. If a resource is not an SMS client, no inventory is collected, so there is no information about that resource in Resource Explorer.

Viewing Hardware Inventory

You can find the hardware inventory information collected for a client within the Hardware folder in Resource Explorer. The Hardware History folder contains inventory data that has changed since the previous inventory cycle. These histories remain until you delete the information manually or by using a database maintenance task, such as the **Delete Aged Inventory History** or **Delete Aged Discovery Data** tasks.

The **Hardware** folder contains a wealth of information ranging from specifics about the manufacturer and type of hardware internals to the free space available on each disk. You can use this information to determine which computers to distribute software to, for example, or when to perform remote troubleshooting.

Note

There might be some delay between the collection of hardware inventory data and its appearance in Resource Explorer, depending on where the client is in relation to the SMS site server that Resource Explorer is using, and network or SMS Sender delays.

To view an SMS client's hardware inventory with Resource Explorer, navigate to a collection containing the client in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Collections
   └ collection containing client
```

In the details pane, right-click the client whose information you want to view, point to **All Tasks**, and then click **Start Resource Explorer**. A new window for Resource Explorer opens and displays information about the selected client. Hardware inventory data is under the **Hardware** node.

You can also open Resource Explorer from queries in the SMS Administrator console. The properties returned by the queries must include the resource identifier and resource type. In the details pane, right-click the client whose information you want to view, point to **All Tasks**, and then click **Start Resource Explorer**. A new window for Resource Explorer opens and displays information about the selected client.

SMS keeps historical hardware inventory records for the number of days you specify in the **Delete Aged Inventory History** site maintenance task. For a complete description of this and other database maintenance tasks, see Chapter 13, "Maintaining and Monitoring SMS Systems."

Note

If you double-click a row in the results pane of the Resource Explorer, a properties dialog box is displayed. This dialog box gives a vertical list of the properties and values for that row. This view might be easier to read than the horizontal list in the results pane.

Viewing Hardware Inventory History

To view an SMS client's hardware inventory history with Resource Explorer, navigate to a collection containing the client. In the details pane, right-click the client whose information you want to view, point to **All Tasks**, and then click **Start Resource Explorer**. A new window for Resource Explorer opens and displays information about the selected client. The hardware inventory data is under the **Hardware History** node. Nodes for each date and time that inventory was run are under nodes for the inventory classes that are configured to keep historical data. The most recent data is under the **Current** node.

Data that has not changed does not have a node under **Hardware History**, because there is no history to display.

Viewing Software Inventory

The Resource Explorer **Software** folder contains information collected by software inventory about each type of program file. Resource Explorer displays as much of the following information for each client as could be gathered:

- File name
- File description (if this information was stored for this file)
- File version (if this information was stored for this file)
- File size (measured in bytes)
- File path
- Modified date
- Manufacturer name
- Product name
- Product version
- Product language

In Resource Explorer, information about files whose product details have been collected are listed under the manufacturer's name that developed the software in the **Product Details** folder, and information about files without product details are listed in the **File Details** folder.

To view the inventory of the client's software products that you selected when you configured the Software Inventory Client Agent, start Resource Explorer, double-click **Software**, and then click **Product Details**. The client's software inventory appears in the details pane. If you want to view the inventory of files not associated with products (such as .vbs files), click **File Details**. The inventory of files without product details that are associated with the client appear in the details pane.

Note

Software inventory does not have history. It indicates only the current state of files found on the clients. Files that were inventoried for the client at one time but were later deleted do not appear in the list.

Viewing Collected Files

If file collection is configured in software inventory, the Resource Explorer Software folder contains a Collected Files folder that displays information about the collected files.

The information collected for each file includes:

- File name
- File path
- File size
- Modified date
- Collection date

You can view the contents of a collected file by right-clicking the file name and selecting **View File** from the **All Tasks** menu. You can save the file to your local disk by right-clicking the file name and selecting **Save** from the **All Tasks** menu.

By default, Resource Explorer displays collected files using Notepad. You can have Resource Explorer display the collected files using another program by adding the string value "Viewer" to the following registry key and setting it to the name of the program you want to be used to view collected files:

HKLM\SOFTWARE\Microsoft\SMS\AdminUI\ResourceExplorer

You must include the path to the program if the program is not available in folders listed in the Resource Explorer user's path environment variable.

Reviewing the Inventory Data

SMS inventory returns a large amount of information about your computers. Much of that information can be found in intuitively named classes. However, some commonly used data might be more difficult to find. Table 3.1 lists some commonly used data and where it can be found in SMS. For more information about commonly used data, see Chapter 3, "Advanced Inventory Collection."

Table 3.1 Inventory Data Type and Classification in SMS

Data	Inventory method	Property	Resource Explorer group	WMI class (for queries)	SQL Server view (for reports)
Computer Name	Hardware Inventory	Name	Computer System	SMS_G_System_COMPUTER_SYSTEM	v_GS_COMPUTER_SYSTEM
Computer role (server, for example)	Hardware inventory	SystemRole	System	SMS_G_System_SYSTEM	v_GS_SYSTEM

(continued)

Table 3.1 Inventory Data Type and Classification in SMS *(continued)*

Data	Inventory method	Property	Resource Explorer group	WMI class (for queries)	SQL Server view (for reports)
Any hardware details (memory size, for example)	Hardware inventory	TotalPhysical Memory	Memory	SMS_G_System_X86_PC_MEMORY	v_GS_X86_PC_MEMORY
Software configuration details (services, for example)	Hardware inventory	DisplayName	Services	SMS_G_System_SERVICE	v_GS_SERVICE
CPU type (such as Itanium)	Hardware inventory	ProcessorType	Processor	SMS_G_System_Processor	v_GS_PROCESSOR
CPU model (such as Pentium IV)	Hardware inventory	Name	Processor	SMS_G_System_Processor	v_GS_PROCESSOR
CPU speed	Hardware inventory	Current_Clock_Speed	Processor	SMS_G_System_Processor	v_GS_PROCESSOR
Operating system	Hardware inventory	Caption	Operating System	SMS_G_System_OPERATING_SYSTEM	v_GS_OPERATING_SYSTEM
SMS client type (Advanced Client vs. Legacy Client)	Discovery	ClientType	Not in the Resource Explorer. Available as a property of the resource.	SMS_R_System	v_R_System
Software installed via Add/Remove Programs	Hardware inventory	All	Add or Remove Programs	SMS_G_System_ADD_REMOVE_PROGRAMS	v_GS_ADD_REMOVE_PROGRAMS
Software inventory product details	Software inventory	All	Product Details	SMS_G_System_SoftwareProduct	v_GS_SoftwareProduct

(continued)

Table 3.1 Inventory Data Type and Classification in SMS *(continued)*

Data	Inventory method	Property	Resource Explorer group	WMI class (for queries)	SQL Server view (for reports)
Software inventory file details if product known	Software inventory	All	Product Details	SMS_G_System_SoftwareFile	v_GS_SoftwareFile
Software inventory file details if product not known	Software inventory	All	File Details	SMS_G_System_UnknownFile	v_GS_UnknownFile
Software inventory collected files	Software inventory	All	Collected Files	SMS_G_System_CollectedFile	v_GS_CollectedFile
Last software inventory collection date and time	Software inventory	LastScanDate	Last Software Scan	SMS_G_System_LastSoftwareScan	v_GS_LastSoftwareScan
Last file collection date and time	Software inventory	LastCollectedFileScanDate	Last Software Scan	SMS_G_System_LastSoftwareScan	v_GS_LastSoftwareScan
Last hardware inventory collection date and time	Hardware inventory	LastHardwareScan	Workstation Status	SMS_G_System_WORKSTATION_STATUS	v_GS_WORKSTATION_STATUS
Hardware history	Hardware inventory	All	Hardware History	SMS_GH_System_*	v_HS_*
NOIDMIF details	Hardware inventory	All	Group name from the MIF	SMS_G_System_ + the group class from the MIF	v_GS_ + the group class from the MIF

(continued)

Table 3.1 Inventory Data Type and Classification in SMS *(continued)*

Data	Inventory method	Property	Resource Explorer group	WMI class (for queries)	SQL Server view (for reports)
IDMIF details	Hardware inventory	All	Not applicable. Resource Explorer does not display non-system resources.	SMS_G_ + architecture name	v_Gn_ + the group class from the MIF, where n is the architecture number (as recorded in the ArchitectureMap table)
MOF details	Hardware inventory	All	SMS_Group_Name property in the reporting class definition	SMS_G_System_ + the second part of the SMS_Class_ID property in the reporting class definition	v_ GS_ + the second part of the SMS_Class_ID property in the reporting class definition

Any time included in inventory data is the local time at the client, without correction for differences in the time zones or daylight saving time between the server and the client.

The Add or Remove Programs class or view can contain more items than **Add or Remove Programs** in Control Panel. This is because some items are marked as not being able to be removed with **Add or Remove Programs**, so they are not displayed to the users.

Note

In some unusual cases, SMS might report values for properties, such as CPU type, that are not accurate. In most cases, SMS obtains the values from WMI. So in the case of CPU type, this might be due to the fact that the CPU type is newer than the version of WMI that you are running. Updating WMI (by updating the operating system, possibly with a service pack) might correct the inaccuracy. When first developing a report or other feature that depends on inventory data, you should review the data closely to ensure that no such issues apply to the data you are using.

Other Considerations for Collecting Inventory

Some special scenarios apply to software and hardware inventory. You should be aware of these scenarios in case they apply to your SMS clients.

Hardware and Software Inventory Behavior When Clients Cannot Connect to the SMS Site

SMS clients might not always be able to connect to a CAP or a management point, such as when no CAPs or management points are available.

If an SMS client cannot connect to its assigned site, it continues to run hardware and software inventory as configured. The inventory data is collected on the client until a connection is reestablished with a client access point or management point. Remember that inventory data collected after the first inventory include changes in the inventory only. So those outstanding inventories are usually neither large nor redundant.

Collection of User Context Information

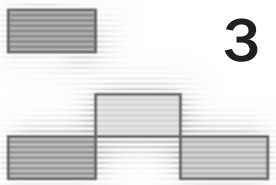
When the Hardware Inventory Agent runs on clients, it runs in the context of the local system account. The agent queries WMI for required data using that context. In some cases (such as environment variables), WMI returns data for all user profiles defined on the computer. In other cases (for example, any file or print shares the user has connected to), WMI returns data for the context in which the data is requested, as opposed to the currently logged-on user. Data collected by hardware inventory might not include the details you expected it to collect. In the example of file and print shares, SMS hardware inventory does not include the user's share connections, because the hardware inventory agent does not run in the user account's context.

A similar issue exists when software inventory encounters encrypted files. Because software inventory is not running in the user's context, files that can be decrypted only by the user cannot be inventoried by SMS. Encrypted files can only have product details inventoried and are collected by SMS when the local system account (or a group that contains the local system account) is given administrative rights to the files.

You can work around this issue by writing a script to store the desired data, and then run that script in the user's context. The script could be run as an SMS advertised program. Using a hardware inventory extension, you can configure hardware inventory to collect that data. For more information about hardware inventory extensions, see Chapter 3, "Advanced Inventory Collection."

CHAPTER 3

Advanced Inventory Collection



The topics described in Chapter 2, “Collecting Hardware and Software Inventory,” provide sufficient information for you to use hardware and software inventory effectively. However, you can enhance Microsoft® Systems Management Server (SMS) inventory functionality with two techniques described in this chapter.

In This Chapter

- Using Resource Explorer from the Command Line
- Extending Hardware Inventory

Using Resource Explorer from the Command Line

Usually, you run Resource Explorer from the SMS 2003 Administrator console. You can also run it from the command line by specifying one of the following:

- An explicit resource using the resource identifier
- A query that returns a resource

When using Resource Explorer from the command line, you might also need to specify a collection that the resource belongs to, for example, if the user does not have appropriate security credentials to access all resources, but has credentials for accessing specific collections.

Using Resource Explorer from the command line is frequently a faster way to view data than using the SMS Administrator console for occasional inventory data review.

Specifying an Explicit Resource

Use the following syntax to specify an explicit resource to display in Resource Explorer.

```
mmc explore.msc -s -sms:ResourceID=n -sms:Connection=<namespace path>
```

where:

- *n* is the ResourceID of the SMS client that you want to display inventory for.
- <*namespace path*> is the path to the Windows Management Instrumentation (WMI) namespace that contains the SMS client data.

For example, the following command displays inventory data for the client associated with ResourceID=1:

```
mmc c:\sms\bin\i386\explore.msc -s -sms:ResourceID=1 -
sms:Connection=\\<MyServer>\root\sms\<SMS_site code>
```

Using a Query to Specify a Resource

Use the following syntax to specify a query that returns a resource to display in Resource Explorer.

```
mmc explore.msc -s -sms:ResExplrQuery=<WQL Query> -sms:Connection=<namespace
path>
```

where:

- <*WQL Query*> is a valid WMI Query Language (WQL) query that returns the ResourceID of the SMS client that you want to display inventory for.
- <*namespace path*> is the path to the WMI namespace that contains the SMS client data.

For example, the following command opens Resource Explorer with inventory data for the client named “MyComputer” that belongs to the SMS site “ABC” having a primary site server named “MyServer”:

```
mmc c:\sms\bin\i386\explore.msc -s -sms:ResExplrQuery="SELECT ResourceID FROM
SMS_R_SYSTEM WHERE Name = "'MyComputer'" -
sms:connection=\\MyServer\root\sms\site_ABC
```

Your query might return more than one instance, but Resource Explorer uses only the first instance that is returned.

Using a Collection

Using Resource Explorer from the command line enforces the same security as using Resource Explorer from the SMS Administrator console. If you do not have **Read Resource collections** class rights to view the resource, you must specify a collection that grants you the proper credentials to view the resource.

Use the following syntax to specify the resource to display in Resource Explorer.

```
mmc explore.msc -s -sms:CollectionID=<Collection ID> -sms:ResourceID=n -
sms:Connection=<namespace path>
```

-Or-

```
mmc explore.msc -s -sms:CollectionID=<Collection ID> -
sms:ResExplrQuery=<WQL Query> -sms:Connection=<namespace path>
```

where:

- <*Collection ID*> identifies the collection that the resource belongs to, such as SMS00001.
- *n* is the ResourceID of the SMS client that you want to display inventory data for.
- <*namespace path*> is the path to the WMI namespace that contains the SMS client data.
- <*WQL Query*> is a valid WQL query that returns a ResourceID of the SMS client that you want to display inventory data for.

Extending Hardware Inventory

If you want to extend SMS hardware inventory, WMI provides data in a large number of classes that are not defined in SMS_def.mof. You can also create special classes of your own.

Note

Because SMS hardware inventory can collect details about the software on your computers, you can think of the hardware inventory extension options as also giving you the option to extend software inventory, although the extensions do not affect the software inventory subsystem itself.

Creating Hardware Inventory Extensions

You can use either of the following ways to extend SMS hardware inventory:

- Using Management Information Format (MIF)-based extensions
- Using Managed Object Format (MOF)-based extensions

Also, you can write scripts that dynamically create either MIF or MOF extensions.

MIF extensions are based on an older standard than MOF standards. MIF extensions are less flexible than MOF extensions, and do not provide the benefits that WMI provides. MIF extensions are most appropriate for relatively static data. MOF extensions are appropriate for both static and dynamic data. MOF extensions are generally preferred, but if you already have a MIF-based extension, or if you find MIFs simpler, then you might choose to use MIF extensions.

The one thing that MIF extensions can do that MOF extensions cannot do is to create new architectures, such as new types of resources, and data for those architectures. However, you can also define new architectures by using custom discovery data records (DDRs). For information about on how to create new architectures using DDRs, see Appendix C, "Scripting SMS Operations."

Hardware inventory extensions are collected at the same time that normal hardware inventory is collected. If you want to start hardware inventory on demand (for testing purposes, for example), see Chapter 4, "Understanding SMS Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Propagating Hardware Inventory Extensions Throughout the SMS Hierarchy

If you are using hardware inventory extensions only for queries, reporting, or reviewing computer status with Resource Explorer, you can implement the extensions in any part of your SMS hierarchy that you want. However, if you create query-based collections that reference hardware inventory extension classes, you should implement those extensions at the SMS site where the collections are created, and at all its lower level sites. The extensions do not need to be implemented at all clients at those sites; one client is sufficient. You can use the SMS site server itself as that client.

Because all collections are automatically propagated to all child sites, extensions must be implemented at all lower level sites of the site where the collections are created. If the collections that are dependent on the extension classes cannot find those classes, a status message is generated frequently at all sites, which can consume network bandwidth.

To address this issue, you can create a package that copies your hardware inventory extension into place on the site servers. Then, create a query-based collection for SMS site servers and advertise the package to that collection. In the future, if you add a site server, it automatically becomes a member of the collection and receives the hardware inventory extension.

MIF Extensions

MIF is part of the Desktop Management industry standard. The MIF standard defines how text files can be used to represent computer management information. Because MIF is an industry standard, programs that store management data in MIF files do not need to be SMS-specific. However, SMS can collect the MIFs and store them in the SMS site database, where you can use their data in the same ways that you use default SMS inventory data.

You can also create MIF files by using a text editor. When you have defined a MIF file that stores the data you require, you can use that file as a template so that similar data is defined in the same manner. For example, when you are setting up a new computer, you can copy the template file to the new computer, edit the data contained within the file to reflect the new computer, and then save the new file. SMS collects the file and stores the information in the SMS site database, along with the other inventory data for that computer.

Your MIF file might contain information about a user's phone number, job title, office number, and similar details that SMS cannot automatically determine.

For SMS, standard MIF files are called NOIDMIF files. These files do not contain a unique identifier for the data. They have no ID. SMS automatically associates NOIDMIF file data with the computer that the NOIDMIF files are collected from.

SMS also supports IDMIF MIF files. These files do contain a unique ID, and are not associated with the computer they are collected from. IDMIF files can be used to collect inventory data about devices that are in the vicinity of a computer, but not actually associated with it. For example, a shared network printer, video cassette recorder, photocopier, or similar equipment is not associated with any specific computer, but you might want to record data about it for asset management purposes. This data is stored in separate tables in the SMS site database.

IDMIF extensions (or custom DDRs) can also be used to create new tables in the SMS site database that you might need for reporting purposes. For example, you might have asset management data that is not strongly tied to individual computers. This data is not appropriate for NOIDMIF files or MOF extensions, but you want to join it with SMS data for reporting purposes.

Caution

Removing IDMIF extensions from clients does not cause the associated data to be removed from the SMS site servers.

Customizing with NOIDMIF Files

NOIDMIF files must be stored in the following folder on Advanced Clients:

%*Windir*%\System32\CCM\Inventory\Noidmifs

NOIDMIF files must be stored in the following folder on Legacy Clients:

%*Windir*%\MS\SMS\Noidmifs

The safest method on both clients is to use the folder that the following registry subkey points to:

HKLM\Software\Microsoft\SMS\Client\Configuration\Client Properties\ NOIDMIF Directory

If the classes defined in the NOIDMIF files do not already exist on the primary site server, the site server's Inventory Data Loader creates the new classes on the existing architectures. After that, inventory for that client includes the new classes by processing the NOIDMIF file each time inventory is run. For example, if a NOIDMIF file creates a class called **Asset Number**, that custom MIF file causes the Inventory Data Loader to create the class **Asset Number**. Each time inventory is run, the Hardware Inventory Client Agent processes the NOIDMIF file again and replaces any values that have changed. If the NOIDMIF file is removed from the destination folder, all the classes and properties are deleted the next time hardware inventory runs, except from the history.

► To customize a single client by using a NOIDMIF file

1. Prepare the NOIDMIF file by performing the steps listed in the "To create a NOIDMIF file to add the Wide World Asset Numbers class" procedure later in this section.
2. Place the NOIDMIF file in the NOIDMIF folder. For example, on a Legacy Client:

```
copy test.mif %windir%\MS\SMS\Noidmifs
```

The next time hardware inventory runs, the NOIDMIF file is included in the process, and the new properties and classes are added to the SMS site database.

Creating a Class by Using a NOIDMIF File

The most common way to use a NOIDMIF file is to create a new class that cannot be collected with inventory, and then store it in the SMS site database. For example, before SMS was installed on their network, Wide World Importers catalogued each computer in the organization by using a company-assigned asset number. These numbers were assigned and collected by hand. With SMS, administrators from Wide World Importers can use a NOIDMIF file to add the asset number for each client computer to its other information within the SMS site database, so that it is available for queries and asset management. Because the asset number is then associated with collected inventory properties, much more information is always available to administrators. The following sample NOIDMIF file illustrates this process:

```
Start Component
   Name = "System Information"
   Start Group
      Name  = "Wide World Asset Numbers"
      ID    = 1
      Class = "wideWorldAssetNumbers"
      Key = 1
```

(continued)

(continued)

```
      Start Attribute
         Name    = "Computer Asset Number"
         ID      = 1
         Type    = String(10)
         Value   = "414207"
      End Attribute
   End Group
End Component
```

Note

The value is stored as a string because, in some reporting tools, commas are automatically inserted for integer values, which can cause the format of the asset number to change.

You can create NOIDMIF files by using the MIFgen tool included in the Microsoft BackOffice® 4.5 Resource Kit, or you can create them by using any text editor.

To create such a NOIDMIF file using a text editor, use the following procedure.

To create a NOIDMIF file to add the Wide World Importers Asset Numbers class

1. Type the following line to begin the NOIDMIF file:

   ```
   Start Component
   ```

   You must always add a component and name the component when you create a NOIDMIF file.

2. Type the following line to name the component:

   ```
   Name = "System Information"
   ```

   By using a general name such as **System Information**, this component becomes more flexible. You can then use it to add any information you want to maintain for this client by adding new groups to the existing NOIDMIF file.

3. Type the following line to add the Display Name for the new **Wide World Importers Asset Numbers** class:

   ```
   Start Group

   Name = "Wide World Importers Asset Numbers"
   ```

   The **Name** property is the string that administrators see in Resource Explorer to refer to this class. Wide World Importers Asset Numbers is a DMTF group class. When SMS first loads this group, it creates a WMI class called SMS_G_wide_world_asset_numbers.

   After you add properties, even if you add only a single property, you need to add a group to contain your new properties.

4. Type the following line to give the **Wide World Importers Asset Numbers** class a group ID number:

   ```
   ID = 1
   ```

   Use any method to determine the unique ID number for each group and property, if the ID number is unique for groups within this component.

5. Type the following line to add the **wideWorldImportersAssetNumbers** class:

   ```
   Class = "wideWorldImportersAssetNumbers"
   ```

   The **Class** information is used for processing and is never seen by administrators.

6. Type the following line to add the key property:

   ```
   Key = 1
   ```

   This entry indicates that the first property listed is the key. Key properties are unique properties that identify instances of a certain class. Whenever you have more than one instance of a class, you must include at least one key property, or the subsequent instances of the class overwrite the previous instances. If no key properties are defined for a NOIDMIF file on a client running a 32-bit operating system, all the properties are designated as key by the inventory process. This does not occur for IDMIF files or for NOIDMIF files on clients running 16-bit operating systems.

7. Type the following lines to add the first property:

   ```
   Start Attribute

   Name = "Computer Asset Number"

   ID = 1

   Type = String(10)

   Value = "414207"

   End Attribute
   ```

   You must set an ID number for this property, name the property, and then specify a data type. The ID number you choose must be unique within the group. Only three data types are recognized by the system: integer, string, and specially formatted **DateTime** string. You must also specify a valid value for the data type you selected.

When you use a NOIDMIF file to define a new class, the class is inventoried at the next cycle, because the NOIDMIF file is processed on the client.

When you customize hardware inventory by using NOIDMIF files, you must leave the NOIDMIF in the NOIDMIFS folder on the client. The custom MIF file is used at each hardware inventory cycle when the extended classes and properties are collected. If the NOIDMIF file is not found on the client during hardware inventory, the extended classes and properties are deleted and you must submit the NOIDMIF file again by replacing it in the NOIDMIFS folder on the client.

The NOIDMIF file in this example is manually created and its values are static. The values are updated only when someone edits the file. SMS hardware inventory then collects the updated file and updates the corresponding data in the SMS site database.

Customizing with IDMIF Files

You can use IDMIF files to create entire new architectures in the SMS site database, or to update existing architectures. They can also be used to add stand-alone computers to the SMS site database. IDMIF files are also frequently used to inventory non-system items.

IDMIF files are identical to NOIDMIF files, with these exceptions:

- IDMIF files must have a delta header that provides architecture, and a unique ID. NOIDMIF files are automatically given a similar header by the system during processing on the client.
- IDMIF files must include a top-level group with the same class as the architecture being added or changed, and that group must include at least one property.
- Like NOIDMIF files, IDMIF files have key properties that must be unique. Any class that has more than one instance must have at least one key property defined, or subsequent instances overwrite previous instances.

Requirements of IDMIF Files

Two delta header comments are required for an IDMIF file. Other comments are optional. The comments you must include are:

- The name of the architecture you want to create or modify:

  //Architecture<*ArchitectureName*>

- A unique ID for this instance:

  //UniqueID<*UniqueID*>

The unique ID can be any unique ID. Each architecture has one or more instances within the SMS site database. The unique ID is the key for this specific instance.

Also, although it is not required, you should use the agent name, especially with a large or complicated custom MIF file that might be updated by more than one agent.

//AgentID<*AgentName*>

If you do not include this attribute, hardware inventory might overwrite the information your IDMIF file places in the SMS site database.

The agent name enables you to independently create and modify the **System** architecture. Others who modify the architecture can use a different agent name. They can then remove or modify the parts of the architecture that are associated with that agent, independently of the modifications of other agents.

There is another requirement of any IDMIF file. Whenever you create an IDMIF file, you must include a group within the IDMIF file with the same class name as the architecture you are creating or modifying. This group is known as the top-level group.

Also, if you create any class that has more than one instance, you must include at least one key value within the class, to avoid having each instance overwrite previous instances.

Important

The formatting of the comments must be exactly the same as that given here. The only part that you can change is the part in italics. The **<** and **>** characters must be included.

IDMIF files must be stored in the following folder on Advanced Clients:

%*Windir*%\System32\CCM\Inventory\Idmifs

IDMIF files must be stored in the following folder on Legacy Clients:

%*Windir*%\MS\SMS\Idmifs

The safest method on both clients is to use the folder the following registry key points to:

HKLM\Software\Microsoft\SMS\Client\Configuration\Client Properties\IDMIF Directory

The following is an example of a simple IDMIF file:

```
//Architecture<Widget>
//UniqueId<414207>
Start Component
   Name = "System Information"
   Start Group
      Name  = "Widget Group"
      ID    = 1
      Class = "Widget"
      Key = 1
      Start Attribute
         Name    = "Widget Asset Number"
         ID      = 1
         Type    = String(10)
         Value   = "414207"
      End Attribute
   End Group
End Component
```

MOF Extensions

Management Object Format (MOF) is part of the Web-based Enterprise Management (WBEM) industry standard. The Microsoft implementation of WBEM is called Windows Management Instrumentation (WMI). The MOF standard defines how text files can be used to represent computer management information, objects that define computer management information, and related structures.

Because WBEM is an industry standard, programs that store management data in WBEM, which is implemented as WMI in Microsoft Windows® operating systems, do not need to be SMS-specific. However, SMS can collect the WMI data and store it in the SMS site database where you can use the data in the same ways that you use default SMS inventory data.

Understanding the Relationship Between the Hardware Inventory Agent and WMI

Understanding the relationship between the SMS Hardware Inventory Client Agent and WMI is important to understand the classes that must be defined in MOF extensions to hardware inventory. This understanding must be based on a knowledge of WMI. For an introduction to WMI, see Appendix B, "Window Management Instrumentation."

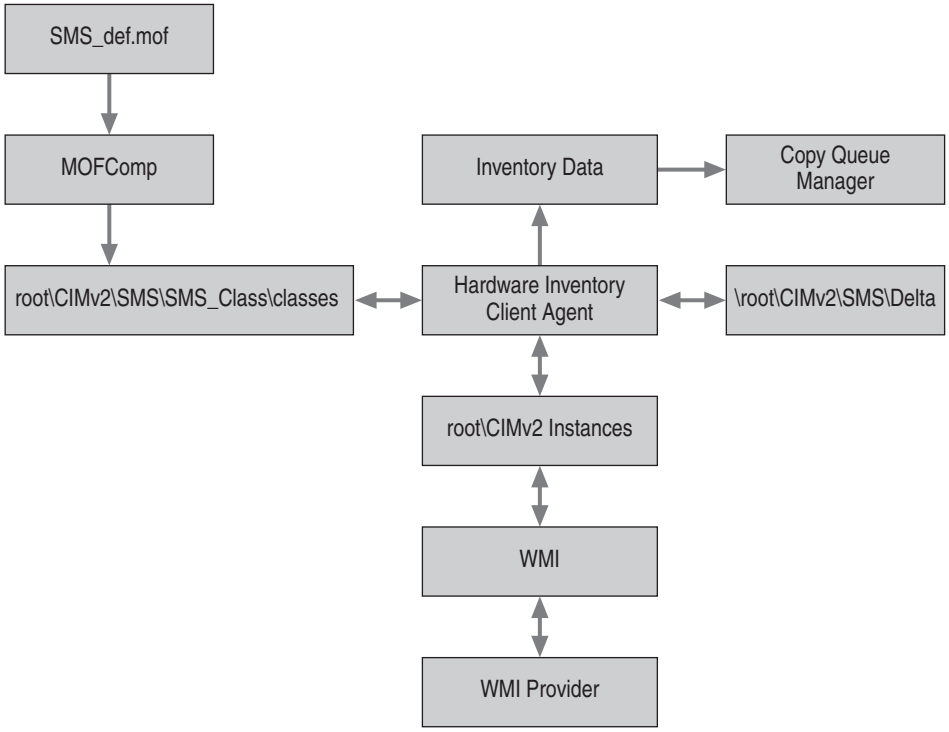
By default, the SMS Hardware Inventory Client Agent retrieves data from the WMI CIMv2 namespace. The agent does not retrieve all the data from the CIMv2 namespace. Instead, it retrieves specific data based on hardware inventory rules stored in the CCM\policy\machine\actualConfig namespace on the Advanced Client and the CIMv2\SMS namespace on the Legacy Client. The Advanced Client stores the rules as instances in the InventoryDataItem class. The Legacy Client stores the rules as qualifiers on classes that mirror the classes in the CIMv2 namespace.

The hardware inventory rules are defined in the SMS_def.mof file, as described in "Configuring Hardware Inventory Rules" section in Chapter 2, "Collecting Hardware and Software Inventory." The SMS_def.mof file provided on the SMS site server is automatically propagated to all SMS clients and automatically compiled on those clients. For Advanced Clients, the SMS_def.mof is changed into Advanced Client policy that is made available to the Advanced Clients. For Legacy Clients, the SMS_def.mof is propagated in its native form and compiled on the SMS clients. The compilation of SMS_def.mof places the hardware inventory rules in the SMS_def.mof into the CIMv2\SMS namespace.

The classes in the CIMv2 namespace are called data classes because they contain the data that the Hardware Inventory Client Agent collects. The instances in the Advanced Client CCM\policy\machine\actualConfig namespace are called reporting instances because those classes instruct the Hardware Inventory Client Agent as to which data classes and properties should be collected and then reported to the SMS site. The classes in the Legacy Client CIMv2\SMS namespace are called the reporting classes.

Figure 3.1 illustrates the relationships among the namespaces used by the Legacy Client hardware inventory agent.

Figure 3.1 The relationships among the SMS hardware inventory namespaces and the Legacy Client hardware inventory agent



Changes to the SMS_def.mof file are propagated to all SMS clients (both Advanced and Legacy Clients) by way of the normal Legacy Client maintenance components of SMS. When the Hardware Inventory Client Agent runs, it checks whether the SMS_def.mof file has changed on the Legacy Client. If so, it uses MOFComp.exe to compile the SMS_def.mof into the root\CIMv2\SMS namespace, under the SMS_Class superclass.

The Hardware Inventory Client Agent then scans the root\CIMv2\SMS namespace for classes that are flagged to be reported, and looks in the \root\CIMv2 namespace for classes with the same name. WMI provides the instances for those classes, often by using WMI Providers that work with the underlying systems, such as the operating system, to provide the data. If providers are not used to provide the data, the data is statically defined as instances for the classes. Statically defined instances are updated by scripts or programs, or by compiling MOF files.

Note

The Hardware Inventory Client Agent does not look for data classes in the \root\CIMv2 namespace in these two scenarios:

- If the class has the SMS_Namespace qualifier set to true
- If the Namespace qualifier has been used

Only Microsoft uses the SMS_Namespace qualifier. For more information about the Namespace qualifier, see the "Using MOF Extensions with Namespaces Other Than root\CIMv2" section later in this chapter.

The Hardware Inventory Client Agent compares the collected data with the data in the \root\CIMv2\SMS\Delta namespace to determine what data has changed and therefore should be reported. If a full inventory is requested, as with a resynchronization request, all the collected data is reported.

The inventory data is then provided to the Legacy Client's copy queue manager, which uploads the data to a client access point (CAP) at each of the client's assigned sites (if they have hardware inventory enabled). For the Advanced Client, inventory data is sent up the SMS hierarchy to the assigned management point.

Customizing with MOF Files

MOF files are appropriate for static management data or dynamic management data. Static data includes details such as the computer user's phone number, office number, and name. Dynamic data includes details such as Microsoft SQL Server™ database sizes and applications installed with Windows Installer.

Using MOF Extensions for Static Data

You can create MOF files by using a text editor. When you have defined a MOF file that stores the data you require, you can use that MOF file as a template so that similar data is defined in the same manner. For example, when you are setting up a new computer, you could copy the template file to the new computer, edit the data contained within the file to reflect the new computer, and then save and compile the new file. Compiling the MOF places the data in WMI. SMS can then collect the data from WMI and store the information in the SMS site database along with the other inventory data for that computer.

MOFs that store static data must do two things:

1. Define the data class.
2. Define the data (instances), as in this example:

```
#pragma namespace ("\\\\.\\root\\CIMv2")
class Static_MOF
{
     [key]
        string user;
        string office;
        string phone_number;
};

instance of Static_MOF
{
        user = "John Smith";
        office = "Building 4, Room 26";
        phone_number = "(425) 707-9791";
};

instance of Static_MOF
{
        user = "Denise Smith";
        office = "Building 4, Room 26";
        phone_number = "(425) 707-9790";
};
```

After you edit the MOF file on the client computer to enter the data, the file must be compiled by using the Mofcomp.exe command, as in this example:

```
Mofcomp.exe <path>\SMS_def.mof
```

You can edit and compile the file repeatedly, but because it is a manual process, you might not want to use this process for data that changes frequently.

Also, SMS_def.mof must be extended to include a reporting class for the collected data. For example, add the following MOF to SMS_def.mof:

```
#pragma namespace ("\\\\.\\root\\CIMv2\\sms")
[ SMS_Report     (TRUE),
  SMS_Group_Name ("Static AssetInfo MOF"),
  SMS_Class_ID   ("MICROSOFT|Static_MOF|1.0")]

class Static_MOF : SMS_Class_Template
{
    [SMS_Report(TRUE), key]
        string user;
    [SMS_Report(TRUE)]
        string office;
    [SMS_Report(TRUE)]
        string phone_number;
};
```

Using MOF Extensions for Dynamic Data

MOF extensions for dynamic data are much like MOF extensions for static data, except that they do not include the data itself. Instead, they provide details for WMI to retrieve the data using WMI providers. For more information about WMI providers, see Appendix B, "Windows Management Instrumentation," and the *Microsoft Windows Management Instrumentation Software Development Kit*, which is available for download at http://msdn.microsoft.com.

You can create MOF files with details for WMI to retrieve data by using a text editor. Compiling the MOF places the hardware inventory rules in WMI. SMS can then collect the data from WMI based on the hardware inventory rules and store the information in the SMS site database along with the other inventory data for that computer.

You can add MOFs that are used to collect dynamic data to SMS_def.mof. The reporting class part of the MOF must be added to SMS_def.mof. The data class part of the MOF can be added to SMS_def.mof for Legacy Clients. For Advanced Clients, the data class part of the MOF must be distributed to the clients and compiled using the WMI MOFcomp.exe tool. You can use SMS software distribution to do this. If the data class uses a WMI provider that is not standard on the clients, the WMI provider must also be distributed to all clients.

MOFs that provide hardware inventory rules for dynamic data must do two things:

1. Define any providers the data class might require, if the providers are not already defined in the MOF file
2. Define the data class

Also, SMS_def.mof must be extended to include a reporting class for the collected data.

After you edit the MOF file to enter the data, the file must be compiled using the MOFcomp.exe tool. You can edit and compile the MOF file repeatedly, but because the data is automatically collected, you would do this only to correct errors with the MOF.

The examples in the "Common MOF Extensions" section later in this chapter are all examples of MOF extensions for dynamic data. Adjusting an example to serve your needs might be easier than reading the relevant WMI SDK documentation.

Using MOF Extensions with Namespaces Other Than root\CIMv2

The SMS Hardware Inventory Client Agent typically collects data from the root\CIMv2 namespace. Data that you want hardware inventory to collect might be located in other namespaces. This is often true for systems that have their own WMI providers, such as SQL Server, Microsoft Exchange, and Microsoft Internet Information Services.

The Hardware Inventory Client Agent on the Legacy Client cannot access namespaces other than root\CIMv2. However, the WMI View Provider can be used to make data from those namespaces available in the root\CIMv2 namespace. For information about using the View Provider, see the WMI SDK. For an example of using the View Provider, see the "Collecting SQL Server Information" section later in this chapter.

The Hardware Inventory Client Agent on the Advanced Client can access namespaces other than root\CIMv2 by using a reporting class qualifier. When defining your MOF extensions, add the Namespace qualifier to your hardware inventory rules. The following example demonstrates using the Namespace qualifier:

```
            #pragma namespace ("\\\\.\\root\\CIMv2\\sms")
[SMS_Report(TRUE),
    SMS_Group_Name("Registered GUIDs"),
    SMS_Class_ID("Microsoft|Registered GUIDs|1.0"),
    Namespace("\\\\\\\\.\\\\root\\\\WMI")]
class RegisteredGuids : SMS_Class_Template
{
    [SMS_Report(TRUE), key]
    string InstanceName;
    [SMS_Report(TRUE)]
    boolean Active;
    [SMS_Report(TRUE)]
    uint32 GuidType;
    [SMS_Report(TRUE)]
    uint32 LoggerId;
    [SMS_Report(TRUE)]
    uint32 EnableLevel;
    [SMS_Report(TRUE)]
    uint32 EnableFlags;
    [SMS_Report(TRUE)]
    boolean IsEnabled;
};
```

Best Practices for MOF Extensions

Here are some best practices for extending SMS hardware inventory using MOFs:

- Back up your current MOF file before making changes to it.
- After you add your own classes to SMS_def.mof, do not use MOF Manager to further customize SMS_def.mof. All further customizations, including enabling and disabling the reporting of classes or properties, should then be done by editing the file with a text editor.
- If you add your MOF to SMS_def.mof, you should add your MOF to the end of SMS_def.mof. This minimizes the possibility of your extensions interfering with the hardware inventory rules that Microsoft supplies.
- Ensure that data hardware inventory rules always compile into the root\CIMv2 namespace and the reporting hardware inventory rules compile into the root\CIMv2\SMS namespace. The #pragma namespace lines define which namespace the following lines compile into, so their placement is important.
- The class name for the reporting class must be identical to the class name of the data class.
- The reporting class must have all the same key properties as the data class. The other properties do not need to be included in the reporting class. However, any properties that are included must have the same data type in both the data and reporting classes.

- The reporting class must be based on the SMS_Class_Template class. The " SMS_Class_Template" clause, as illustrated in the example MOFs, ensures this.
- Providers must be defined only once in a MOF. If you merge MOFs, remove redundant hardware inventory rules.
- When creating reporting hardware inventory rules, consider using the data class definition as a starting point. Use Wbemdump.exe or MOF Generator in CIM Studio to export the data class definition to a MOF file. Both of these tools are included in the Windows Management Instrumentation SDK. Then edit that MOF file to put the class in the CIMv2\SMS namespace and add in the qualifiers that SMS requires. Use SMS_def.mof as your source for examples.
- Test MOF extensions on individual clients in a lab environment before deploying more broadly. This testing allows you to ensure the MOF accomplishes exactly what you want. You should watch to ensure that the MOF does not return too much data. However, the reporting class changes must be added to the site-wide SMS_def.mof. Otherwise, the site does not load the data. Your testing should be done in your test lab before being deployed on any clients in the production environment.
- For clients that fail to return data for the extension you create, use Wbemtest.exe or CIM Studio, as described in Appendix B, "Windows Management Instrumentation," to ensure that the data class contains instances. If the data class does not contain instances but should contain extensions, correct the problem with the data class part of your extension.
- On the Legacy Client, review the Hinv32.log on any clients that fail to return data for your hardware inventory extension. A "CLASS - Process Class:" line should be listed for your extension, and there should be no error messages related to your class after it. If you do see error messages, correct the problem with the reporting class part of your extension.

  On the Advanced Client, review the Inventoryagent.log file. In particular, look at the "Inventory: Query =" lines.
- The data class you create does not have any SMS-specific requirements. Create the data class by using the documentation for the provider that provides the class data, the WMI SDK, and any other WMI documentation.
- The WMI registry provider has three variations, as instance, property, and event providers. Use the variant that is appropriate for your requirement. The Power_Mgmt MOF in the "Finding Computers That Are Laptops" section later in this chapter is an example of a registry property provider MOF. The Hotfixes MOF in the "Finding Hotfix Information" section later in this chapter is an example of a registry instances provider.

  The registry instances provider is appropriate when you need to collect an unpredictable but consistently formatted set of registry values under a predetermined registry key. But most registry entries do not fit this description. They are trees of keys that have predictable names and inconsistent data types or names. For more information about the WMI registry provider, see the WMI SDK.
- Ensure that all reporting classes are included in the SMS_def.mof. Data for reporting classes that are only defined at the Advanced Clients is ignored at the site server.

Scripted Extensions

Some details are difficult or impossible to collect using MIF or MOF hardware inventory extensions. In those cases, consider writing a script to collect the details using any of the many techniques available to script, and then add the details to the SMS hardware inventory.

Scripts can write static or dynamic MIF or MOF files. Scripts that write MIF files use exactly the same techniques as any script that writes text files. Those techniques are well documented in many sources, so this chapter does not describe how to write scripts that write MIF files.

If a script writes to a MOF file, the MOF file then has to be compiled, so it is more efficient to write the MOF data directly to WMI. The rest of this section describes how to write scripts that write to WMI. The WMI principles are the same as those described in the "Common MOF Extensions" section later in this chapter.

Scripts that write hardware inventory extension data to WMI must do three things:

1. Create the data class, if it does not exist already.
2. Collect the data.
3. Write the data to WMI.

In addition, SMS_def.mof must be extended to include a reporting class for the collected data. For example, add the following MOF to SMS_def.mof:

```
#pragma namespace("\\\\.\\ROOT\\CIMV2\\sms")
[SMS_ReporT(TRUE),
SMS_Group_Name("Asset Wizard Results"),
SMS_Class_ID("MICROSOFT|ASSETWIZARD|1.0")]

class SMS_AssetWizard_1 : SMS_Class_Template
{
     [SMS_Report(TRUE),key] uint32 Type;
     [SMS_Report(TRUE)] string ContactFullName;
     [SMS_Report(TRUE)] string ContactEmail;
     [SMS_Report(TRUE)] string ContactPhone;
     [SMS_Report(TRUE)] string ContactLocation;
     [SMS_Report(TRUE)] string SysLocationSite;
     [SMS_Report(TRUE)] string SysLocationBuilding;
     [SMS_Report(TRUE)] string SysLocationRoom;
     [SMS_Report(TRUE)] string SysUnitManufacturer;
     [SMS_Report(TRUE)] string SysUnitModel;
     [SMS_Report(TRUE)] string SysUnitAssetNumber;
     [SMS_Report(TRUE)] boolean SysUnitIsLaptop;
};
```

The *Microsoft Systems Management Server 2003 Software Development Kit* includes a Visual Basic program, Asset Wizard, which prompts the user for various details, such as the user's office number and telephone number. It then adds the details to the SMS hardware inventory. The next example adds the same details to the SMS hardware inventory, but from a script. The example illustrates all the steps to write to WMI except for collecting the data. In this example, the data is in the script itself. You can use any technique to collect the data that is supported by scripting.

```
Set loc = CreateObject("WbemScripting.SWbemLocator")
Set WbemServices = loc.ConnectServer(, "root\CIMv2")

    On Error Resume Next
    Set WbemObject = WbemServices.Get("SMS_AssetWizard_1")
'If this call failed, we need to make the SMS_AssetWizard_1 data class
If Err Then
   'Retrieve blank class
   Set WbemObject = WbemServices.Get
   'Set class name
   WbemObject.Path_.Class = "SMS_AssetWizard_1"
   'Add Properties (8 = CIM_STRING, 11 = CIM_BOOLEAN)
   WbemObject.Properties_.Add "Type", 19
   WbemObject.Properties_.Add "ContactFullName", 8
   WbemObject.Properties_.Add "ContactEmail", 8
   WbemObject.Properties_.Add "ContactPhone", 8
   WbemObject.Properties_.Add "ContactLocation", 8
   WbemObject.Properties_.Add "SysLocationSite", 8
   WbemObject.Properties_.Add "SysLocationBuilding", 8
   WbemObject.Properties_.Add "SysLocationRoom", 8
   WbemObject.Properties_.Add "SysUnitManufacturer", 8
   WbemObject.Properties_.Add "SysUnitModel", 8
   WbemObject.Properties_.Add "SysUnitAssetNumber", 8
   WbemObject.Properties_.Add "SysUnitIsLaptop", 11

   'Add key qualifier to Type property
   WbemObject.Properties_("Type").Qualifiers_.Add "key", True
   WbemObject.Put_
End if
On Error Goto 0

Set WbemServices = loc.ConnectServer(, "root\CIMv2")
Set WbemObject = WbemServices.Get("SMS_AssetWizard_1").SpawnInstance_
' Store property values (the data!)
WbemObject.Type = 0
WbemObject.ContactFullName = "John Smith"
WbemObject.ContactEmail = "JSmith"
WbemObject.ContactPhone = "(425) 707-9791"
WbemObject.ContactLocation = "Redmond"
WbemObject.SysLocationSite = "Campus"
```

(continued)

(continued)

```
WbemObject.SysLocationBuilding = "24"
WbemObject.SysLocationRoom = "1168"
WbemObject.SysUnitManufacturer = "Dell"
WbemObject.SysUnitModel = "GX1"
WbemObject.SysUnitAssetNumber = "357701"
WbemObject.SysUnitIsLaptop = False
'WMI will overwrite the existing instance
WbemObject.Put_
```

Changing or Removing Hardware Inventory Extensions

When you implement hardware inventory extensions, new classes and tables are created in the following locations:

- WMI data and reporting classes on the SMS clients. This is true unless you used MIFs, which do not use WMI on Legacy Clients and have no WMI data and reporting classes. The Advanced Client has reporting policies instead of reporting classes, but they serve the same purpose.
- Tables in SQL Server on the SMS sites that the clients report to (or the site's parent site, if the client is assigned to a secondary site), and their higher level sites.
- SQL Server views on each of the client's higher level sites.
- WMI classes in the SMS site namespace of the client's higher level sites.

If you remove a hardware inventory extension, you might want to remove these entries.

To remove the client-side classes, remove the reporting hardware inventory rules from SMS_def.mof and use the deleteclass pragma to remove the data and reporting classes on the clients like this:

```
#pragma namespace("\\\\.\\root\\CIMv2")
#pragma deleteclass("Static_MOF", NOFAIL)
```

Caution

Do not remove the data class if your hardware inventory extension did not create it. Do not remove the data class data if the data is dynamic and can be deleted. (If the provider, such as the Registry provider, does not support deletion, your attempt to delete the data is ignored.)

```
#pragma namespace("\\\\.\\root\\CIMv2\\sms")
#pragma deleteclass("Static_MOF", NOFAIL)
```

If you have only Advanced Clients in your SMS hierarchy, you can remove the reporting class by removing it from the SMS_def.mof at each SMS site. SMS automatically removes the relevant reporting policies from the Advanced Clients, so the classes are no longer reported.

To remove the tables on the SQL Servers, use Delgrp.exe from the *Microsoft BackOffice 4.5 Resource Kit* on each of the primary sites. To remove the tables on many site servers, you can distribute the Delgrp.exe tool (with appropriate parameters) by using SMS software distribution.

An example of a command using Delgrp.exe is:

```
Delgrp "MICROSOFT|STATIC_MOF|1.0"
```

The server-side classes are automatically removed as soon as the SQL Server tables are removed.

You can make changes to a hardware inventory extension by removing the previous extension, and then implementing the extension with the changes. You can also make changes without removing the previous extension, but if any data has been collected with the previous extension, the new extension causes new class and table names to be created. The old data is purged by the SMS site database maintenance tasks, but in the meantime, both sets of data are available, possibly causing confusion.

Common MOF Extensions

You can extend SMS hardware inventory by using MOFs in as many ways as WMI can be extended. However, some MOF extensions are particularly popular because they help deliver solutions for common computer management needs.

Finding Computers That Are Laptops

Determining which computers are laptops is useful in a variety of circumstances. For example, you might want to install the Advanced Client only on laptops. If all of your computers are already discovered and inventoried by SMS, you can create a collection for the laptops and then advertise the Advanced Client to the laptops. However, computer vendors do not use a standardized method to identify laptops. Consider the alternatives and use whichever methods are appropriate for the laptops in your organization.

To identify laptops, consider using the following hardware inventory properties:

- Win32_SystemEnclosure, ChassisTypes(1)=10. This property when set to the value of 10 is equivalent to “notebook.” However, not all computers provide this property. This class is defined in the SMS_def.mof.
- Win32_Battery or Win32_PortableBattery. If any instances exist, then the computer is probably a laptop. However, uninterruptible power supplies sometimes are reported as batteries, so this might not be reliable if some of your computers have uninterruptible power supplies. This class is defined in the SMS_def.mof, but reporting is not enabled by default.
- Win32_PCMCIAController. If any instances exist, the computer is probably a laptop. This class is defined in the SMS_def.mof but reporting is not enabled by default.
- Win32_DriverVXD.Name = “pccard”. If any instances exist, the computer is probably a laptop. However, this option works only on Microsoft Windows 98 computers. This class and property are enabled for reporting by default.

- Win32_ComputerSystem.Manufacturer. If you purchase your laptops from a different vendor than your desktop computer and server vendor, this value might reliably identify your laptops. This class and property are enabled for reporting by default.
- Win32_ComputerSystem.Model. You might need to check for a variety of different models to include all of your laptops. This class and property are enabled for reporting by default.
- Static record. You could define your own property in a MIF or MOF and set it when the computer is originally set up for use in the production environment.
- Power scheme. Laptops usually use the Portable/Laptop power scheme (number 1). This is a registry entry, so you can use the following MOF to collect power scheme data, which uses the WMI property registry provider:

```
#pragma namespace("\\\\.\\root\\CIMv2")
// Registry property provider
instance of __Win32Provider as $PropProv
{
    Name    ="RegPropProv" ;
    ClsID   = "{72967901-68EC-11d0-B729-00AA0062CBB7}";
    ImpersonationLevel = 1;
    PerUserInitialization = "FALSE";
};

instance of __PropertyProviderRegistration
{
    Provider    =$PropProv;
    SupportsPut =TRUE;
    SupportsGet =TRUE;
};

[DYNPROPS]
class Power_Mgmt
{
 [key]
 string  index = "current";
 sint32 CurrentPowerPolicy;

};

[DYNPROPS]
instance of Power_Mgmt
{
  [PropertyContext("local|HKEY_CURRENT_USER\\Control
Panel\\PowerCfg|CurrentPowerPolicy"),
        Dynamic, Provider("RegPropProv")]
        CurrentPowerPolicy;
};
```

Note

If you have only Legacy Clients you can include the previous MOF directly in the SMS_def.mof. In this scenario, remove the registry provider definition because it is already defined in SMS_def.mof.

In addition, the following MOF must be added to SMS_def.mof:

```
#pragma namespace ("\\\\.\\root\\CIMv2\\sms")
 [ SMS_Report     (TRUE),
  SMS_Group_Name ("Power Management"),
  SMS_Class_ID   ("MICROSOFT|POWER_MGMT|1.0") ]

class Power_Mgmt : SMS_Class_Template
{
    [SMS_Report(TRUE),key]
    string  index;
    [SMS_Report(TRUE)]
    sint32   CurrentPowerPolicy;
};
```

Finding Computer Serial Numbers

Computer serial numbers are often determined from the BIOS class. However, if you have computers that do not have the serial number available in the BIOS class, try using the system enclosure class, which is enabled by default in SMS_def.mof. If neither class works for your computers, check with the hardware vendor to see if the vendor has a WMI provider, or a program that produces MIFs that include the serial number.

If none of these options work, you must create a MIF or MOF file with the serial number statically recorded. The serial number must be manually entered in that file for each computer. Ensure that the file is preserved (or recreated) if the hard drive is reformatted.

Finding Hotfix Information

Determining which hotfixes have been applied to computers (especially servers), and verifying that a hotfix has been applied to all appropriate computers, are two very important computer management tasks.

Many Windows hotfix installations are recorded in the registry. SMS collects the values from those registry keys using the following MOF:

```
#pragma namespace("\\\\.\\root\\CIMv2")
// Instance provider
instance of __Win32Provider as $InstProv
{
    Name    = "RegProv" ;
    ClsId   = "{fe9af5c0-d3b6-11ce-a5b6-00aa00680c3f}" ;
};

instance of __InstanceProviderRegistration
{
    Provider = $InstProv;
    SupportsPut = TRUE;
    SupportsGet = TRUE;
    SupportsDelete = FALSE;
    SupportsEnumeration = TRUE;
};

[dynamic, provider("RegProv"),
 ClassContext("local|HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows
NT\\CurrentVersion\\Hotfix")
]
class HotFixes
{
  [key]
  string  QNumber;

  [PropertyContext("Installed")]
  uint32 Installed;
};
```

This example demonstrates using the WMI registry instance provider. The Add or Remove Programs example in the SMS_def.mof is also an example that demonstrates using the WMI registry instance provider.

Also, add the following MOF to SMS_def.mof:

```
#pragma namespace("\\\\.\\root\\CIMv2\\sms")
[SMS_Report(TRUE),
SMS_Group_Name("Hotfixes"),
SMS_Class_ID("MICROSOFT|HOTFIXES|1.0")]
class HotFixes : SMS_Class_Template
{
  [SMS_Report(TRUE),key]
  string  QNumber;

  [SMS_Report(TRUE)]
  uint32 Installed;
};
```

Note

Although the example provided in this section applies to hotfixes, you might be able to apply the same methodology to other software and tools released to customers between major software release dates. This includes security patches, critical updates, service packs, and other interim updates. For more information, see your program documentation.

For those hotfixes that do not modify this registry key, you can modify your hotfix installation procedure to add this registry entry.

The registry instance provider is useful when the registry keys you are collecting have:

- A known parent registry key in the registry.
- Consistent value names.
- An unknown number of instances.
- Key names that are not known ahead of time.

Note

This example is included to illustrate the instance version of the WMI Registry Provider. For reporting on hotfixes, consider using comprehensive solutions available from Microsoft, including SMS Feature Packs.

The WMI registry property provider cannot be used to collect such registry values because the registry property provider requires that the key names be known at the time the MOF is created, and that the number of instances is also known. The primary benefit of the WMI registry property provider is that registry entries from different locations in the registry can be combined in the class.

Collecting Windows Installer Information

Another way to check for software that is installed on SMS client computers is to collect details on products that use Windows Installer. Current professional software often has installation procedures based on Windows Installer.

The Windows Installer provider provides many classes and properties, but the following MOF might provide sufficient detail when added to SMS_def.mof:

```
#pragma namespace ("\\\\.\\root\\CIMv2\\sms")

[ SMS_Report     (TRUE),
  SMS_Group_Name ("Windows Installer Installed Products"),
  SMS_Class_ID   ("MICROSOFT|MSI_PRODUCTS|1.0") ]
```

(continued)

(continued)

```
class Win32_Product : SMS_Class_Template
{
    [SMS_Report(TRUE), key]
        string IdentifyingNumber;
    [SMS_Report(TRUE), key]
    string Name;
    [SMS_Report(TRUE)]
        string Vendor;
    [SMS_Report(TRUE), key]
        string Version;
    [SMS_Report(TRUE)]
        string PackageCache;
    [SMS_Report(TRUE)]
        string InstallDate;
    [SMS_Report(TRUE)]
        string InstallLocation;
};
```

The Windows Installer data classes are predefined in the CIMv2 namespace, so you do not need to define the data class.

Collecting SQL Server Information

Computers running SQL Server 2000 have a WMI provider that you can use to return a rich set of management data for SQL Server. The WMI provider must be installed as described in the SQL Server documentation. SMS collects that data for centralized reporting or management. For example, the following MOF collects information about the databases:

```
#pragma namespace("\\\\.\\Root\\CIMV2")

instance of __Win32Provider as $DataProv
{
    Name = "MS_VIEW_INSTANCE_PROVIDER";
    ClsId = "{AA70DDF4-E11C-11D1-ABB0-00C04FD9159E}";
    ImpersonationLevel = 1;
    PerUserInitialization = "True";
};
instance of __InstanceProviderRegistration
```

(continued)

(continued)

```
{
    Provider = $DataProv;
    SupportsPut = True;
    SupportsGet = True;
    SupportsDelete = True;
    SupportsEnumeration = True;
    QuerySupportLevels = {"WQL:UnarySelect"};
};

[union, ViewSources{"Select * from MSSQL_Database"},
ViewSpaces{"\\\\.\\root\\MicrosoftSQLServer"}, Dynamic : ToInstance,
provider("MS_VIEW_INSTANCE_PROVIDER")]
class SQL_Databases
{
  [PropertySources("Size") ] sint32 Size;
  [PropertySources("SQLServerName"), key ] string SQLServerName;
  [PropertySources("Name"), key ] string Name;
  [PropertySources("SpaceAvailable") ] sint32 SpaceAvailable;
};
```

Also, add the following MOF to SMS_def.mof:

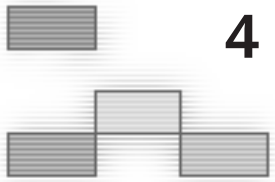
```
#pragma namespace("\\\\.\\root\\CIMv2\\sms")
[SMS_Report(TRUE),
SMS_Group_Name("SQL Database"),
SMS_Class_ID("MICROSOFT|SQLDatabase|1.0")]
class SQL_Databases : SMS_Class_Template
{
  [SMS_Report(TRUE),key]
  string  SQLServerName;
  [SMS_Report(TRUE),key]
  string  Name;
  [SMS_Report(TRUE)]
  sint32 Size;
  [SMS_Report(TRUE)]
  sint32 SpaceAvailable;
};
```

This MOF demonstrates how to collect data from WMI namespaces other than CIMv2 on Legacy Clients. Similar MOFs can collect management information about Microsoft Exchange, Microsoft Office, and many other systems that have WMI providers that populate their own namespaces.

Collecting data from namespaces other than CIMv2 on Legacy Clients is done using the WMI View Provider to create a view class in the CIMv2 namespace based on the class of interest in the other namespace. For more information about the WMI View Provider, see the WMI SDK. For more information about the collecting data from namespaces other than CIMv2 on Advanced Clients, see the “Using MOF Extensions with Namespaces Other Than root\cimv2” section earlier in this chapter.

CHAPTER 4

Managing Collections and Queries



Microsoft® Systems Management Server (SMS) 2003 collections are groups of resources, such as users, user groups, or SMS clients, that have attributes in common. Collections are designed to gather resources into useful groups that you can manage. You can create collections by specifying individual resources. More commonly, you create queries that define targeted resources, and then use the queries to gather resources into a collection. You do this by specifying query-based membership rules for the collection. A query is a specific set of instructions that you use to extract information about a defined set of objects in the SMS site database. If your SMS site uses Active Directory® discovery methods, then you can create queries from Active Directory objects stored in the SMS site database. You can use queries to create collections, but they are also very useful as standalone objects.

Note

All predefined collections and queries that come with SMS 2003 are based on unauthenticated client discovery data, not on inventory data.

Chapter 17, "Discovering Resources and Installing Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide,* introduced the concepts of resources and resource discovery. This chapter describes how to manage your SMS resources using collections and queries.

In This Chapter

- Working with Collections
- Working with Queries

Working with Collections

Collections serve as targets for SMS operations, primarily software distribution. By using collections, you can perform an SMS operation on every member of the collection at the same time. For example, when you want to distribute software to clients with certain minimum hardware requirements, you can use a collection of clients that meet those hardware requirements. A client must be in a collection before you can perform any SMS operation on that client.

SMS includes many predefined collections that are useful in most SMS sites. You can use these collections as they are, or you can customize them. You also can create your own collections. This section lists some of the ways you use collections as you work with SMS. There are three main topics in this section:

- Understanding Collections
- Creating and Managing Collections
- Managing Resources in Collections

Understanding Collections

Collections are sets of resources that are grouped together because they satisfy one or more rules. You can use collections to group resources in a logical order instead of the physical order of groups such as sites. Collections also provide a manageable view into the SMS site database by partitioning the data into useful categories. Collections gather resources according to user-defined criteria.

You define and set membership rules for each collection. Membership rules are the criteria by which SMS determines whether a resource is a member of a particular collection. A membership rule is based on one of the following:

SMS query You can create membership rules based on a query (query rules). The resources returned from the query become members of the collection.

Specific resource or group You can create membership rules that target individual resources, such as a list of users, user groups, or SMS clients (direct rules). The targeted resources become permanent members of the collection. By targeting individual resources, you can gather a diverse group of resources.

Note

When you create a collection based on a query, SMS imports the query statement and stores it along with the other information about the collection. If you subsequently modify the query, the collection is not automatically updated. To update the collection, you must re-import the modified query statement.

After you set the membership rules for a collection, you can use the collection as a target for software distribution and other management tasks. A resource can be a member of as many collections as you think are appropriate. You can define the rules for collections at any time. You do not need to wait until resources are discovered.

Updating collection membership

Collections are dynamic. SMS periodically evaluates resources against the membership rules. When SMS discovers resources, it adds those resources to any collection with membership rules that match the resources. If you modify the membership rules of a collection, the effect on the membership list is reflected the next time the collection is evaluated. You can schedule collection evaluations for a later time, or to recur at a specific interval. You also can update the list of resources on demand.

Note

Updating a collection membership list does not automatically refresh the view of the collection in the details pane of the SMS Administrator console. Instead, an hourglass appears next to the name of the collection in the console tree as a reminder to refresh the view. To refresh the view of an updated collection, select the collection and press F5.

When hardware and software configurations on individual computers change, SMS removes those computers from collections or adds new computers to collections according to the membership rules of the collections. By keeping collections current, SMS ensures that your software distributions always go to all the computers that meet your collection criteria, including those computers that were added to the network after you created the collection. In a similar manner, if a computer no longer meets the criteria for a collection, then it no longer receives software targeted to that collection. For example, if a computer is moved to a different group or no longer has the minimum free disk space specified in the collection criteria, it might be removed from the collection.

Understanding collection changes in SMS 2003

Predefined collections remain relatively unchanged in SMS 2003 from SMS 2.0. However, the underlying SMS 2003 database structure has been updated to accommodate new database objects such as Active Directory objects. Some, but not all, predefined collections display Active Directory objects. For example, the **All User Groups** collection in SMS 2003 contains data obtained only from Windows User Group Discovery to maintain interoperability with SMS 2.0. The collection does not contain Active Directory System Group Discovery or Active Directory User Discovery data.

Note

Some predefined collections and queries found in SMS 2.0 are not present in SMS 2003.

Collections That Provide Management Scope

SMS collections are meant to reflect how your organization commonly organizes users, user groups, and computers for software distributions and other tasks. Sites are organized by the geography of your organization, and collections are organized into logical groups.

Many organizations find it necessary to have more than one department within the company managed by the same SMS site. For example, at Northwind Traders, the marketing department, the sales department, and the human resources department are all in the same physical location. However, the IT department might determine that it is best to have one SMS site containing the marketing, sales, and human resources departments, but the administration of each department handled by the department itself. The IT department decides to:

- Create a central site containing all three departments.
- Create three collections in the central site, one including clients from the marketing department, one including clients from the sales department, and one including clients from the human resources department.
- Install an SMS Administrator console in each department.
- Give the IT employees in each department the security rights to manage their respective collections.

In this way, Northwind Traders can group their clients and servers by physical location in a manner that is most efficient for their network. At the same time, by creating collections that match their management structure, they can allow the administration to be based on logical rules instead of physical location. They also increase the security of each department by organizing them in this way.

Subcollections

In addition to resources, collections can contain other collections, which are called subcollections. Subcollections are not members of the containing collection. A collection can be a subcollection of multiple collections. This is important because it means that multiple instances of a collection can appear throughout the hierarchy. This also means that you can delete one instance of a collection and still have other instances of that same collection appear elsewhere as subcollections.

Subcollections do not inherit the attributes of the parent collection. Membership rules of collections and subcollections are completely separate. The query that creates a collection is completely separate from the query that creates the subcollection. Subcollections function in the same way as nested distribution lists within an e-mail system. The nested distribution list has its own identity and is simply a convenient way of gathering the diverse set of groups that form the distribution list. In the same way, subcollections are a convenient way to gather several diverse groups of resources into a single group to be acted on in some way.

Any operation that you can perform on a collection you can also perform on its subcollections. If collection A contains collection B as a subcollection, then operations that you performed on collection A also can be performed on collection B. For example, software advertised to collection A also can be advertised to collection B, and to any subcollections of collection B.

You can create a subcollection in two ways:

- By creating a new collection under an existing collection.
- By linking a collection to another existing collection.

Singularly dependent subcollections

If you create a new collection under an existing collection, that subcollection is singularly dependent on the collection under which it was created, as long as you do not link other collections to it. When you delete a collection, any singularly dependent subcollections of that collection are also deleted. Any advertisements, queries, or collection membership rules that are dependent on the subcollection are impacted by its deletion. You might want to use the Collection Deletion Wizard to delete singularly dependent subcollections before you delete the collection on which they are dependent. For more information, see the "Deleting a Collection" section later in this chapter.

Multiple dependent subcollections

If you create a new subcollection under an existing collection, and then link other collections to that subcollection, then the subcollection becomes dependent on multiple collections. When you delete a collection, multiple dependent subcollections are not deleted if they are still subcollections of the remaining collections that link to it. This remains the case until all but one of the linking collections has been deleted. Then, the subcollection becomes singularly dependent on the remaining collection.

Collections in the SMS Hierarchy

When you create a collection at a parent site, SMS propagates it to child sites, which can be either primary or secondary sites. You cannot modify these propagated collections at a child site. SMS uses a special icon for these propagated collections to signal that they are locked and cannot be modified.

When SMS propagates a collection, primary child sites receive all the data about a collection, including general data, membership rules, and a list of subcollections, but they do not receive the actual resource list for the collection. Each primary child site generates a resource list for its own site. There are two advantages to having the primary child site generate its own resource list—the transmission from SMS is smaller, and the resource list is kept up-to-date more easily.

Note

When you create a linked collection at a child site by specifying a collection propagated from a parent site, the linked collection cannot be removed at the child site because it is locked. However, you can delete the linked collection at the parent site, which also deletes all instances of the collection at the child site.

It is possible for you to add new resource classes on a parent site and not add those same resource classes on its child sites. You then can create a collection on the parent site with membership rules that define resources within the extended resource classes. When such collections are propagated down to a child site that does not also contain the extended resource classes, the collection still runs. It returns all resources defined by the membership rules for resource classes that are found on the child site. However, because such collections contain membership rules that are not evaluated by the child site, SMS generates a detailed status message for each such rule and a milestone status message at the end of the collection evaluation. These messages are generated only once per day for each such collection.

Secondary child sites receive the list of collection members that belong to their secondary sites, but they do not receive membership rules because they do not maintain a site database. When a primary site collection is re-evaluated, the primary site sends updated membership lists to its secondary sites to replace outdated lists.

Collection and Resource Security

In SMS, you maintain security by creating security rights that specify the permissions that a user or user group has for various SMS security objects — collections, packages, advertisements, and status objects. You can create a security right for an entire class of objects, such as all collections, or just for specific instances, which are individual collections. You might have a requirement to restrict the permissions of some administrators to work with only a specific group of resources. You can do this by creating a collection or collections that contain the targeted resources, and then granting permissions so that the administrators can manage only the specific collection or collections.

For example, suppose that your organization has collections named Engineering, Human Resources, and Finance. If a system administrator manages the resources only for the Engineering department, you can grant that administrator permission for only that collection. There is no need to grant permissions to that administrator for the other collections. The system administrator can perform SMS operations on the Engineering collection, but not on the other collections.

Unlike other SMS objects, you can also grant permissions for the resources in a collection, including **Delete Resource**, **Modify Resource**, **Read Resource**, **Use Remote Tools**, and **View Collected Files**. When you grant resource permissions, it is for all resources in a particular collection, not for individual resources. For example, if a user has **Delete Resource** permission for collection A, the user can delete any of the resources in collection A.

It is important to note that if you grant permissions to a user for resources in a collection, the permissions extend to the same resources contained in other collections. This is regardless of the permissions that the user has for the other collections. For example, if you grant a user **Modify Resource** permission for the **All Windows 98 Systems** collection, that user can modify clients running Microsoft Windows® 98 contained in any collection.

For more information about SMS security, see Chapter 5, “Understanding SMS Security,” in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Collection Limiting

Collection limiting is a method of restricting the scope of a query or a collection membership rule. A query that is limited to a collection only returns resources that are in the specified collection, even if other resources in the SMS site database match the query criteria.

While collection limiting can be used to filter query results, it is most often used as part of resource security. You might have a requirement to limit the permissions of some administrators to work with only a specific group of resources. You can do this by creating a collection or collections that contain the targeted resources, and then specifying the permissions so that the administrators can manage only a specific collection or collections.

In previous versions of SMS, to view instances of a secured resource, a user had to limit to a collection for which they had instance-level **Read** permission. To view inventory, or inventory history, a user had to limit to a collection for which they had **Read Resource** permission. If the user did not specify collection limiting, they did not see any results.

SMS 2003 uses automatic collection limiting. Although you can still explicitly specify collection limiting, if you do not, then SMS 2003 limits the resources that are returned to members of all collections for which the user has appropriate rights. If a user queries against resources and collection limiting is not specified, then the user sees only those resources that are members of collections to which the user has **Read** permission. If a user queries against inventory data, the user sees only the inventory for resources that belong to collections to which the user has **Read Resource** permission.

Creating and Managing Collections

You must have the appropriate permissions for the Collection security object class to create, export, or import a collection. You must also have the appropriate permissions for the Collection security object class or instance to modify, delete the collection, or to view the properties of resources in a collection. For more information about permissions, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

To create a new collection

1. Navigate to **Collections** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Collections
```

2. Right-click **Collections**, point to **New**, and then click **Collection**.
3. In the **Collection Properties** dialog box, use the tabs to complete the property settings for your new collection.

For more information about creating a new collection, see the SMS Help.

Note

You cannot create a new collection with the same name as an existing collection.

► To modify a collection

1. In the SMS Administrator console, navigate to **Collections**.
2. Right-click a collection, and then click **Properties**.
3. In the **<*Collection name*> Collection Properties** dialog box, change the appropriate properties.

For more information about creating a new collection, see the SMS Help.

If you modify membership rules, SMS prompts you to update the resource list of the collection. If you target a collection for an advertisement, and then subsequently modify the membership rules for that collection, it affects the software distribution to the clients in that collection. Clients that are removed from the collection do not receive the advertisement. New clients do receive the advertisement.

Creating Subcollections

By creating subcollections, you can include or exclude the subcollections in a given operation on the collection. For example, when you create an advertisement that specifies a collection that has subcollections, you can decide whether or not to distribute to each of the subcollections.

You can create a subcollection in two ways:

- By linking the collection to another existing collection
- By creating a new collection under an existing collection

► To create a subcollection by linking to another collection

1. In the SMS Administrator console, navigate to **Collections**.
2. Right-click the collection for which you want to create a subcollection, point to **New**, and then click **Link to Collection**.
3. In the **Browse Collection** dialog box, select the collection that you want to add as a subcollection.

► To create a subcollection by creating a new collection

1. In the SMS Administrator console, navigate to **Collections**.
2. Right-click the collection for which you want to create a subcollection, point to **New**, and then click **Collection**.
3. In the **Collection Properties** dialog box, use the tabs to complete the property settings for your new collection.

Note

After you create subcollections, when you view **Collections** in the SMS Administrator console tree, the same collection name appears in more than one place. In each instance, the name refers to the same collection.

Deleting a Collection

You can delete collections by using the SMS Delete Collection Wizard; however, when you delete a collection:

- Resources in the collection are not deleted from the SMS site database.
- SMS administrators whose security rights are limited to the resources in the deleted collection can no longer view those resources.
- Advertisements to the collection are deleted.
- Queries and query-based membership rules that are limited to the collection are no longer limited. Queries that are no longer limited to collections do not prompt you for a limiting collection when run.
- Singularly dependent subcollections of the collection are deleted. For more information, see the "Subcollections" section earlier in this chapter.

Note

A collection can be a subcollection of multiple collections. This is important because it means that multiple instances of a collection can appear throughout the hierarchy. If you delete one instance of a collection, other instances of that collection might still appear elsewhere as subcollections.

To start the Collection Deletion Wizard

1. In the SMS Administrator console, navigate to **Collections**.
2. Right-click the collection that you want to delete, and then click **Delete**.

The wizard cautions you about the effects of deleting a collection and provides information about the objects listed earlier in this section.

Exporting or Importing Collections

You can use the Export Object Wizard and the Import Object Wizard to export or import SMS collections. When you export a collection, the collection's definitions are written to a Managed Object Format (MOF) file, which is a text file that can be imported. You cannot transfer a collection with direct membership rules from one site to another.

You must have **Read** permission for the Collections security object class or instance to export a collection. You must have **Create** permission for the Collections security object class to import collections.

When a collection is exported as a MOF file, the collection's Object ID is not written to the MOF file. This prevents an existing collection from being accidentally replaced if you import a MOF file and the Object ID of an imported collection matches the Object ID of an existing collection.

When you import collections, ensure that none of the collections have the same name as an existing collection. If you do so, the data for the existing collection is replaced without warning. To change the name of a collection in a MOF file, you can open and edit the MOF file with any text editor.

Importing multiple object classes

You can use the Export Object Wizard to export objects from only one object class that includes reports, collections, or queries at a time. MOF files that are created by using the Export Object Wizard contain only one object class. You can use the Import Object Wizard to import user-created MOF files that contain objects from multiple object classes. However, if you do not have **Create** permission for all object classes in a MOF file, some objects might not be imported. For example, if a MOF file contains both reports and collections and you have **Create** permission only for the Reports object class, the collections are not imported.

To export collections

1. In the SMS Administrator console, navigate to **Collections** and right-click **Collections**.

   –Or–

   In the SMS Administrator console, navigate to **Collections** and right-click the collection that you want to export.
2. Point to **All Tasks**, and then click **Export Objects**.
3. Complete the Export Object Wizard, and then click **Finish**.

For more information about completing the Export Object Wizard, see the SMS Help.

To import collections

1. In the SMS Administrator console, navigate to **Site Database**.
2. Right-click **Site Database**, point to **All Tasks**, and then click **Import Objects**.
3. Complete the Import Object Wizard, and then click **Finish**.

For more information about completing the Import Object Wizard, see the SMS Help.

Caution

Do not import a collection with a name that is the same as the name of an existing collection. If you do so, the properties of the existing collection are replaced without warning. To avoid this, you can open the MOF file by using any text file application and check the object names against the name of existing objects in the SMS site database.

Managing Resources in Collections

In SMS, a resource is any object, such as a client, a user, or a user group, that can be discovered and potentially managed by SMS. You can gather resources into collections to better manage the resources in your site.

Updating a Collection Resource List

When you create a collection, SMS adds all resources that fit the membership rules you have specified for the collection. When SMS adds a new resource to the SMS site database, it also adds the resource to any collections that apply the next time those collections are updated. You can configure a collection to be automatically updated according to a specified schedule. You can also update a collection's resource list on demand.

For predefined collections and each new collection that you create, the default update schedule is every day. To increase site performance:

- Increase or eliminate the update schedule period, if appropriate.
- Delete unnecessary collections.

When you update a collection on demand, the resource list for the collection is updated, and SMS also sends the collection's definition down to any child sites to be updated. Updating all collections on demand might decrease system performance during the process.

► To modify the recurring update schedule for a collection

1. In the SMS Administrator console, navigate to **Collections**.
2. Right-click a collection and click **Properties**.
3. In the **Properties** dialog box, click the **Membership Rules** tab, and then click **Schedule**.
4. In the **Schedule** dialog box, specify when and how often you want to update the collection.

► To update the resource lists of all collections on demand

1. In the SMS Administrator console, navigate to **Collections**.
2. Right-click **Collections**, point to **All Tasks**, and then click **Update Collection Membership**.

► To limit the number of resources displayed in collections

1. In the SMS Administrator console, navigate to **Collections**.
2. Right-click **Collections**, and then click **Properties**.

   The **Collections Properties** dialog box opens.
3. On the **General** tab, select the **Limit number of collection members** check box.
4. In the **Limit** box, specify the maximum number of resources for each collection to display in the details pane.

Note

To display all resources for each collection in the details pane, enter **0** in the **Limit** box.

Deleting a Resource

Sometimes resources are no longer needed in collections, and it might be useful to delete them.

Caution

When you delete a resource from a collection, all information about the resource is removed from the SMS site database, including all discovery, inventory, and history data. The resource is also deleted from all other collections that it is a member of. This results in the client being unmanaged. Advanced Client policy is not removed, so Advanced Clients might continue running SMS tasks and might report status to their assigned management point.

A deleted resource might be rediscovered and, if it still meets the membership rules, be added back to the collection.

To delete a resource

1. In the SMS Administrator console, navigate to **Collections**.
2. Double-click the collection containing the resource you want to delete.
3. Right-click the resource and click **Delete**.
4. In the **Confirm Delete** dialog box, click **Yes** to confirm the deletion of the resource.

Deleting All Resources in a Collection

You can also delete all resources in a collection at one time.

Note

If the deleted collection is large, and if the resources still exist and are rediscovered, this could take some time and might decrease system performance during the process.

To delete all resources in a collection

1. In the SMS Administrator console, navigate to **Collections**.
2. Right-click a collection, and then click **Delete Special**.
3. When the **Confirm Delete Special** message box appears, click **Yes**.

Caution

When you delete a resource from a collection, all information about the resource is removed from the SMS site database, including all discovery, inventory, and history data. The resource is also deleted from all other collections that it is a member of.

Working with Queries

A query is a specific set of criteria that you use to extract information from the SMS site database. SMS queries store the criteria for sets of database objects that you want to find. A query searches the SMS site database for objects that match the query's criteria. Other SMS features, including Reporting, Collections, and Status Message Queries, use queries against objects within the SMS site database. You can also create standalone named queries, which are stored in the SMS site database, and can be run from within the SMS Administrator console. The results that are returned by a named query appear in the details pane of the SMS Administrator console.

Queries can return information about most types of SMS objects, including sites, advertisements, packages, and named queries themselves. Queries are most commonly used to extract information related to users, user groups, discovered resources, and inventory data.

This section provides an overview of the principles of SMS queries and lists some of the ways you use queries as you work with SMS. There are four main topics in this section:

- Understanding SMS Database Classes
- Understanding SMS Queries
- Creating and Managing SMS Queries
- Creating and Editing Query Statements

Understanding SMS Database Classes

When you build an SMS query, you specify the attribute or attributes within an object type, which is also a Windows Management Instrumentation (WMI) class, that the query uses to search the SMS site database. Any database objects that match one or more specified attributes are returned by the query. An object type is a class containing a set of attributes that represent an SMS database object, such as a client, a user, a user group, a package, or an advertisement. The set of attributes for an object type describe the object. Related attributes are grouped together into attribute classes. SMS object types are WMI classes, and SMS attributes are WMI properties. For a list of the SMS object types, see the "SMS Object Types" section later in this chapter.

For more information about SMS object classes, attributes, and properties, see the *Microsoft Systems Management Server 2003 Software Development Kit*.

The SMS SDK is an excellent source for information about the SMS database and its object classes and attributes. To download the SMS SDK, see the MSDN Web site at http://msdn.microsoft.com.

Another way to understand the SMS classes is to browse the underlying WMI classes, as described in Appendix B, "Windows Management Instrumentation."

Most of the queries that you create are based on the discovery class **SMS_R_System** and on the set of inventory classes that begin with **SMS_G_System**. The **SMS_R_System** class contains discovery data for all discovered SMS system resources, such as clients, printers, routers, users, and user groups. This class includes properties (attributes) such as **IPAddress, OperatingSystemNameandVersion**, and **Name** (system name). The set of **SMS_G_System** classes contain inventory data for the same SMS resources, such as the SMS_G_System_LOGICAL_DISK attribute class. This class contains information about a client's logical disk drive, such as **Availability**, **Name**, **FileSystem**, and **FreeSpace**. The **ResourceID** property links the **SMS_R_System** class and the **SMS_G_System** classes.

If you configure hardware inventory on your SMS site, the Hardware Inventory Client Agent gathers information about the hardware on each client. If you configure software inventory, the Software Inventory Client Agent collects information about specific file types and collects the files you specify. SMS passes this information through the client access point (CAP) or management point to the site server and incorporates hardware and software information into the SMS site database. When the data is available, you can use a query to obtain data from the SMS site database about clients that meet certain criteria; for example, all clients that have less than 256 MB of RAM installed.

Viewing attribute data

One of the best ways to write useful queries is to first view the attribute data directly in the SMS site database. This helps you to confirm that the data you require is available and to identify the classes, instances, and attributes to which you must refer in a query to retrieve that data. Appendix B, "Windows Management Instrumentation," provides useful information about tools, such as CIM Studio, that you can use to view the WMI classes.

You can also use Resource Explorer to determine which attributes you need and what the data type of the value should be. For many queries, your object type is **System Resource**, and if hardware inventory was run on your site, you can use Resource Explorer to narrow your search.

► To use Resource Explorer

1. In the SMS Administrator console, navigate to **Collections**.
2. Locate a client that matches the type of computer that you want to query.
3. Right-click the client, point to **All Tasks**, and then click **Start Resource Explorer**.
4. In the Resource Explorer tree, expand the **Hardware** folder.

   The displayed folders represent each attribute class in the **System Resource** object type. For example, in the **Hardware** folder, the **Logical Disk** folder represents the **SMS_G_System_LOGICAL_DISK** class. Click a folder and view the column names across the top of the details pane. These represent the attributes of that attribute class. For example, in the **Logical Disk** folder, the **File System** column represents the **FileSystem** attribute. The values displayed in the details pane are in the correct data type.

Understanding SMS Queries

SMS queries are similar to queries you might use with Microsoft SQL Server™ or other database management systems, but SMS queries are defined in the WMI Query Language (WQL). You do not need to know WQL to build queries, but it is helpful if you are building more complex queries.

In the SMS Administrator console, you can create queries by using the SMS Query Builder. SMS Query Builder is a user interface designed specifically to help you search for the attributes of objects in the SMS site database and use those to build a query.

► To launch the SMS Query Builder

1. In the SMS Administrator console, navigate to **Queries**.
2. Right-click **Queries**, point to **New**, and then click **Query**.
3. In the **Query Properties** dialog box, click **Edit Query Statement** to launch the SMS Query Builder.

   The **Query Statement Properties** dialog box opens.

The **Query Statement Properties** dialog box is one of the dialog boxes that comprise the SMS Query Builder. The **Query Statement Properties** dialog box opens in the **Query Design** view, from which you can use the tabs and command buttons to build a query. You can also build queries by using WQL in the **Query Language** view by clicking **Show Query Language**.

SMS Query Builder has its own specific terminology and requirements. To understand and use the SMS Query Builder, you must become familiar with the concepts described in the next four sections:

- SMS Object Types
- Required SMS Query Elements
- Optional SMS Query Elements
- WMI Query Language

SMS Object Types

An SMS object type is a resource class containing a set of attributes that represent SMS database objects such as clients, users, user groups, packages, or advertisements. Each object type has specific attributes that describe those objects. The attributes are organized into one or more attribute classes. Attribute classes group related attributes within an object type and contain the set of attributes that define the class. For example, the **System Resource** object type contains the attribute class **Processor**, which includes attributes such as **CurrentClockSpeed** and **Manufacturer**. The **Disk** attribute class includes attributes such as **Partitions** and **SCSIBus**. You use the attributes within an attribute class to construct a query.

When you create a query by using the SMS Query Builder, you can use the attributes of only one SMS object type at a time. By default, the **System Resource** object type is selected. You can use the **<unspecified>** object type to query against more than one SMS object type at a time. For more information, see the "Creating Queries Against Multiple SMS Object Types" section later in this chapter.

The following are brief descriptions of SMS object types that are available for building queries:

Advertisement This object type consists of a single attribute class with attributes representing the data in an SMS advertisement. SMS advertisements are used to alert users that software distributions are available.

Package This object type consists of a single attribute class with attributes representing the data in an SMS package. Packages are basic units of software distribution, including programs and the source files required to run them.

Program This object type consists of a single attribute class with attributes representing the data in an SMS program. Programs are software distribution command lines that install the software or that run the program or command.

Site This object type consists of a single attribute class with attributes representing an SMS site object.

Software metering rule This object type consists of a single attribute class with attributes related to product compliance. This object can help you to enforce product compliance by identifying clients that are not in compliance.

System resource This object type consists of many attribute classes that together characterize the discovery and inventory data of a system resource (a networked client). Discovery data consists of a single attribute class called **System**, and the inventory data consists of the other classes of the **System Resource** object type, such as **Logical Disk**.

User group resource This object type consists of a single attribute class representing the discovery data for **User Group** objects.

User resource This object type consists of a single attribute class representing SMS users in an SMS hierarchy.

Unspecified When you do not specify an object type, you can only create a query by using WQL in the **Query Language** view. This can be useful for creating free-form WQL queries to run against classes other than those listed above, or to run against more than one SMS class.

For more information about SMS object classes, attributes, and properties, see the *Microsoft Systems Management Server 2003 Software Development Kit*.

Another way to understand the SMS classes is to browse them, as described in Appendix B, "Windows Management Instrumentation."

You can also create new object types, also called classes, and attributes that you can use for queries. For more information, see Chapter 2, "Collecting Hardware and Software Inventory."

Required SMS Query Elements

You must specify the following elements in each query. They are found in the SMS Query Builder on the **General** tab of the **Query Statement Properties** dialog box or on dialog boxes that open from that tab.

Query name This element is a unique name that identifies the query. The query name appears in **Queries** in the SMS Administrator console.

Object type This element is an SMS database object that defines the scope of the query. You must designate only one object type for each query. By default, SMS selects the **System Resource** object type. Select your object type based on what you are searching for. For example, if you are looking for all clients that have certain attributes, select the **System Resource** object type. For a list of all SMS object types, see the "SMS Object Types" section earlier in this chapter.

Note

Only resource-related object types, such as System Resource, User Resource, and User Group Resource, can be limited to a collection or used to create a query-based membership rule for a collection. For more information about limiting a query to a collection, see the SMS Help.

Attribute class This element is a container object that groups related attributes. The attributes of an object type are organized into one or more attribute classes. The attribute classes that you can select include all attribute classes belonging to the object type for the current query. In the **Select Attribute** dialog box, which is described later in this chapter, you can select from a list of attribute classes for the object type you selected for this query, and then select an attribute of that class.

Attribute This element is the specific property for which the query searches. In the **Select Attribute** dialog box, you can select from the list of attributes for the attribute class you have chosen.

Optional SMS Query Elements

If you choose to refine your query, additional query elements are required. You can use the **Criteria** and **Joins** tabs of the **Query Statement Properties** dialog box to further refine the query.

The optional SMS query elements include:

- SMS criterion types and values.
- SMS relational operators.
- SMS logical operators.
- SMS query order of precedence.
- SMS attribute class joins.

SMS Criterion Types and Values

You can use an SMS criterion type to create an expression that compares a query attribute to a specified value or to another attribute. The criterion type that you select determines what is compared to the query attribute. For example, if you select the **Simple Value** criterion type, SMS compares the attribute to a constant value that you specify. The criterion properties also specify a relational operator, such as "is equal to" or "is at most," that you use to define the comparison. For example, by using the **Free Space** attribute from the **Logical Disk** attribute class and the **Simple Value** criterion type, you can construct the following expression:

```
LogicalDisk.Free Space is greater than '1500'
```

You can use this expression in a query to search for all clients in your site with more than 1.5 GB of free disk space.

The SMS criterion types are:

Null value Compares the query attribute to null or not null.

Simple value Compares the query attribute to a constant value that you specify.

Note

In the **Criterion Properties** dialog box, you can click **Values**, and if a list of values exists for the attribute you chose, that list appears in a dialog box.

Prompted value SMS prompts you for a value when the query is run. You can use this criterion type to create a query for which you can supply a different value each time than you run it, instead of being limited to a single, static value.

Attribute reference Compares the query attribute to another attribute that you specify.

Subselected values Compares the query attribute to the results that are returned by another query, which you browse to specify.

List of values Compares the attribute to a list of constant values that you specify.

When you create a query expression using a criterion type, you compare an attribute that you specify with a value that you select. Constant values must have a data type that is appropriate for the attribute to which it is being compared. A data type defines the format of a value and the possible range of values. For example, the **NetBIOSName** attribute is stored as a string, and the **DiskStorageSize** attribute is stored as a number.

There are four data types that are used by SMS: numerical, string, date/time, and parameterized. Each query attribute stores data by using one of these data types. When specifying query attributes, the criterion value that you can specify depends on the data type of the query attribute. For relational operators that perform LIKE comparisons such as "is like" or "is not like," you can use wildcard characters within the string. For a list of the wildcards and guidelines for specifying the appropriate criterion value for each of the four data types, see the SMS Help.

SMS Relational Operators

SMS relational operators define how an expression's value is compared to the specified attribute. The relational operators that are available depend on the data type of the attribute.

Numerical operators

You must specify a numeral that the query uses to evaluate the expression. If you specify a value that is not numerical, the query fails.

Date and time operators

You must enter a date that the query can use to evaluate the expression. This value must be entered according to the units specified by the date/time operator. For example, if you use the "year is after" operator, you enter the year by using four digits, such as *2003*.

Date and time operators include the numerical operators for date and time, and specific operators for units of time including millisecond, second, minute, hour, day, week, month, and year.

When you write queries by using the SMS Query Builder, you can express the date and time in any valid SQL format. For more information, see the *Microsoft Systems Management Server 2003 Software Development Kit* or SQL Server Books Online.

String Relational Operators

The evaluation of string relational operators depends on the code page you selected when you installed SQL Server. Each code page has its own order of evaluation. For more information, see the SQL Server product documentation.

SMS Logical Operators

In SMS, you can use logical operators to join two expressions within a query. For example, you can join the following expression:

```
Free Space is greater than 1500
```

with this expression:

```
Processor Name is like %Pentium III%
```

The result is a more complex — and more useful — query:

```
Free Space is greater than 1500
and
Processor Name is like %Pentium III%
```

By using this expression within a query, you can search for all clients on your site that have Pentium III processors and free disk space greater than 1.5 GB. This expression is shown as it appears in the **Query Design** view, which is not the same as the WQL statement in the **Query Language** view.

The logical operators permitted in SMS are as follows:

AND This operator joins two expressions and finds all objects that satisfy both of the expressions joined by AND. You can use AND to narrow the list of objects you want to find. For example, you can search for all clients running Microsoft Windows 2000 Professional and that have more than 1.5 GB of free disk space.

OR This operator joins two expressions and finds all objects that satisfy either of the expressions joined by OR. You can use OR to assemble more than one set of objects in a single group. For example, you can search for all clients running Windows 2000 Professional or Windows 2000 Server.

NOT This operator applies to one expression and finds all objects that do not satisfy the expression following the NOT. You can use NOT to narrow the list of objects you want to find. For example, using AND with NOT you can find all clients that have Pentium III processors with 1.5 GB free disk space and do *not* have Windows 2000 Professional installed.

SMS Query Order of Precedence

Before you can obtain the results you want, you must understand the order in which WQL evaluates the logical operators. On the **Criteria** tab of the **Query Statement Properties** dialog box, the expressions are evaluated from top to bottom except for expressions in parentheses, which always come first. In WQL, expressions are evaluated in the following order:

1. Expressions inside parentheses
2. Expressions preceded by NOT
3. Expressions joined by AND
4. Expressions joined by OR

You can group a set of expressions within parentheses to make complex expressions easier to understand or to force a certain order of evaluation. For example, when more than one OR expression occurs within a complex query, use parentheses to indicate which expressions you want evaluated first. For more information about group parentheses, see SMS Help.

SMS Attribute Class Joins

You use attribute class join operations to specify how to combine data from two different attribute classes. When you use an attribute from an attribute class that is not yet in the query, the SMS Query Builder automatically creates a new join for this attribute class. Suitable joins are automatically created when the query is built. Users typically do not need to use the **Joins** tab of the **Query Statement Properties** dialog box. However, there are certain kinds of queries that can only be expressed by manually entering new joins or modifying the ones that are automatically created. The resulting expression allows you to specify how objects from these classes are related. For example, you can use a join to search for all SMS site database items that have had hardware inventory collected. The following expression is a WQL statement shown as it appears in the **Query Language** view.

```
select * from SMS_R_System
inner join
SMS_G_System_SYSTEM
on SMS_R_System.ResourceID = SMS_G_System_SYSTEM.ResourceID
```

There are four types of attribute-class joins:

Inner join Displays only matching results — always used by joins that are created automatically.

Left outer join Displays all results for the base attribute and only the matching results for the join attribute.

Right outer join Displays all results for the join attribute and only the matching results for the base attribute.

Full join Displays all results for both the base attribute and the join attribute.

Important

Join operations are an advanced function of the WQL language. Before configuring or modifying a join operation, be sure you obtain a good working knowledge of WQL syntax for various types of class joins.

WMI Query Language

WQL is part of the WMI standard. You can review WQL statements associated with the predefined queries provided in the SMS Administrator console to learn more about WQL.

To view the WQL query statement associated with a predefined query

1. In the SMS Administrator console, navigate to **Queries**.
2. Right-click a predefined query and click **Properties**.
3. In the **Query Statement Properties** dialog box, click the **General** tab, and then click **Show Query Language**.

   The WQL query statement appears in the **Query Statement** text box.

A complete description of WQL can be found in the Windows Management Instrumentation SDK, which is available for download from the MSDN Web site at http://msdn.microsoft.com.

Creating and Managing SMS Queries

You must have the appropriate permissions for the Queries security object class to create, export, or import a query. You must also have the appropriate permissions for the Queries security object class or instance to modify, delete, or view the results of the query. For more information about SMS security, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Active Directory Object Queries

Unlike Active Directory, SMS does not store Active Directory objects by distinguished name, which identifies the object and its location in a tree. Instead, SMS stores Active Directory objects by relative distinguished name, so that you can locate an object even if the exact distinguished name is unknown or if it has changed. A relative distinguished name uniquely identifies the object within its parent container.

When building queries to gather Active Directory information, query by relative distinguished name. Because you can have duplicate relative distinguished names for Active Directory objects, you might want to build your query in a way that prevents duplicate relative distinguished names from being returned by the query.

The manner in which you create queries that are based on resource properties discovered by Active Directory discovery methods differs from the way you create queries based on other discovery methods because of the way Active Directory objects are stored in the SMS site database. To obtain user information from Active Directory, you must create queries that query the Active Directory object where user accounts are contained, such as an organizational unit or distribution group. For example, when creating a query based on users' membership in a distribution group, use the User_Group_Name property of the User resource type. Specify the distribution group as *<domain>\<displayed distribution group name>*.

► To run or update the results of a previously run query

1. In the SMS Administrator console, navigate to **Queries**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Queries
```

2. Right-click the query that you want to run or update, and then click **Run Query**.

   –Or–

   Select the query and press F5.

   The query results appear in the console details pane.

You also can run a query and limit the number of items that the query returns.

► To limit the number of items that a query returns

1. In the SMS Administrator console, navigate to **Queries**.
2. Right-click the query that you want to run or update, point to **All Tasks**, and then click **Run Query Special**.
3. In the **Run Query Special** dialog box, specify a limit for the number of items you want returned.

Predefined Queries

SMS 2003 includes a set of predefined queries that you can use to accomplish common resource management tasks. For example, the **Systems by Last Logged On User** query locates the systems where a specified user name is the last user logged on.

Status Message Queries

In addition to the predefined queries, SMS 2003 includes a set of special-function Status Message Queries as part of the SMS Status system. The Status Message Queries can assist you in both monitoring and troubleshooting your SMS sites. For example, you might use the **Queries Created, Modified, or Deleted** message status query to identify changes to queries made within a specified time period.

These specialized queries are located in a different section of the SMS Administrator console. To work with Status Message Queries, navigate to **Status Message Queries**.

```
Systems Management Server
└ Site Database <site code - site name>
 └ System Status
  └ Status Message Queries
```

Note

When a site is upgraded to SMS 2003, Legacy Client Status Message Queries replace SMS 2.0 Client Status Message Queries.

For more information about Status Message Queries, see the SMS Help.

Copying a Predefined Query to Create a New Query

Instead of creating an entirely new query, you might want to modify one of the predefined queries to create a new query. If you modify the predefined queries, you lose the original query. Always make a copy of the predefined query to create your modified version from.

To copy a predefined query to create a new query

1. In the SMS Administrator console, navigate to **Queries**.
2. Right-click **Queries**, point to **New**, and then select **Query**.
3. Click **Browse** and select an existing query.
4. Modify the properties and give the query a unique name.

Creating, Modifying, and Deleting a New Query

To create a new query

1. In the SMS Administrator console, navigate to **Queries**.
2. Right-click **Queries**, point to **New**, and then click **Query**.
3. In the **Query Properties** dialog box, use the **General** and **Security** tabs to specify the query properties.
4. To create or edit the query statement properties, click **Edit Query Statement**. For more information about this process, see the "Creating and Editing Query Statements" section later in this chapter.

Note

You cannot create a new query with the same name as an existing query.

For more information about creating queries, see the SMS Help.

► To modify an existing query

1. In the SMS Administrator console, navigate to **Queries**.
2. Right-click the query that you want to modify. In the **Query Properties** dialog box, use the **General** and **Security** tabs to change the properties that you want to modify.

► To delete a query

1. In the SMS Administrator console, navigate to **Queries**.
2. Right-click the query you want to delete and click **Delete**.

Exporting or Importing Queries

You can use the Export Object Wizard and the Import Object Wizard to export or import SMS queries. When you export a query, the query's definitions are written to a MOF file that then can be imported.

You must have **Read** permission for the Queries security object class or instance to export a query. You must have **Create** permission for the Queries security object class to import queries.

When a query is exported as a MOF file, the query's Object ID is not written to the MOF file. This prevents an existing query from being accidentally replaced if the MOF file is imported and the Object ID of the imported query matches the Object ID of an existing query.

The Export Object Wizard cannot maintain references to other objects. If you export a query that is limited to a collection, then that reference is lost and must be reconfigured when the query is imported.

When you import queries, ensure that none of the queries have the same name as an existing query. If you do so, the data for the existing query is replaced without warning. To change the name of a query in a MOF file, you can open and edit the MOF file with any text editor.

Note

To import a MOF file by using the Import Object Wizard, the file must be in the Unicode file format. All MOF files that are exported by the Export Object Wizard are in the Unicode file format.

Importing multiple object classes

You can use the Export Object Wizard to export objects from only one object class at a time. MOF files that are created by using the Export Object Wizard contain only one object class. You can use the Import Object Wizard to import user-created MOF files that contain objects from multiple object classes. However, if you do not have **Create** permission for all object classes in a MOF file, some objects might not be imported. For example, if a MOF file contains both reports and collections, and you have **Create** permission only for the Reports object class, the collections are not imported.

To export queries

1. In the SMS Administrator console, navigate to and right-click **Queries**.

   –Or–

   Navigate to **Queries** and right-click the query that you want to export.
2. Point to **All Tasks** and click **Export Objects**.
3. Complete the Export Object Wizard, and then click **Finish**.

   For more information about completing the Export Object Wizard, see the SMS Help.

To import queries

1. In the SMS Administrator console, navigate to **Site Database**.
2. Right-click **Site Database**, point to **All Tasks**, and then click **Import Objects**.
3. Complete the Import Object Wizard, and then click **Finish**.

   For more information about completing the Import Object Wizard, see the SMS Help.

Caution

Do not import a query with a name that is the same as the name of an existing query. If you do so, the properties of the existing query are replaced without warning. To avoid this, you can open the MOF file by using any text file application and check the object names against the name of existing objects in the SMS site database.

Creating and Editing Query Statements

The processes for creating or editing a query statement are the same. Before you begin creating or editing query statements, read the "Understanding SMS Queries" section earlier in this chapter.

You can create and edit query statements by:

- Using the **Query Statements Properties** dialog box in **Query Design** view and using the command buttons and properties on the **General**, **Criteria**, and **Joins** tabs.
- Using the **Query Statements Properties** dialog box in **Query Language** view and typing a WQL query statement into the **Query Statement** text box.

This section describes how to create and edit query statements by using the **Query Statements Properties** dialog box in Query Design view.

Important

Use the Query Language view only if you have a good working knowledge of WQL. If you enter a query that is not valid (for example, one that is not syntactically correct), you will get an error message. If the query statement that you edit uses features of WQL that are not supported in the Query Design view, you cannot return to the Query Design view. However, you can still save and run the query.

For information about using WQL, see the SMS SDK and the Windows Management Instrumentation SDK, which are available from the MSDN Web site at http://msdn.microsoft.com.

Creating an Example Query

This section describes, in a series of procedures, the steps that are necessary to create an example query statement. The example query returns all clients running Windows 2000 Professional with Pentium III processors and with more than 1.5 GB of free disk space.

To do this, you must create a query to search the **System Resource** object type, and also create two criteria for the query that narrow the search. The first criteria limits the query results to clients with Pentium III processors, as designated by their description of %Pentium III%. The second criteria limits the query results to clients that satisfy the first condition *and* have more than 1.5 GB of free disk space. You further narrow the results of the query by limiting it to the collection that contains all clients running Windows 2000 Professional.

To create a query statement

1. Navigate to **Queries** in the SMS Administrator console.
2. Right-click **Queries**, point to **New**, and then click **Query**.

   The **Query Properties** dialog box opens. For new queries, the **System Resource** object type is selected by default.
3. Click **Edit Query Statement**.

The **Query Statement Properties** dialog box opens.

Configuring properties on the General tab

You use the **General** tab of the **Query Statement Properties** dialog box to specify which attributes you want to display and to specify how to display the data that the query returns when it is run. If you want all attributes for the specified object type to display, leave the **Results** area blank.

To specify attributes to be displayed

1. In the **Results** area, click **New**.
2. In the **Results Properties** dialog box, click **Select**.

   The **Select Attribute** dialog box opens.
3. Select the **Processor** attribute class from the **Attribute class** list.

4. Select the **Name** attribute class from the **Attribute** list and click **OK**.
5. If you want to sort the query results by using this attribute, in the **Sort** list, select **Ascending** or **Descending**.

> **Note**
>
> Sorting and grouping of array attributes are not supported. If you select any of the following array attributes, then the results data cannot be sorted based on those attributes:
>
> - System Resource: Agent Name, Agent Site, Agent Time, IP Addresses, IP Subnets, IPX Addresses, IPX Network Numbers, MAC Addresses, Resource Names, SMS Assigned Sites, SMS Installed Sites, System Roles
> - User Resource: Agent Name, Agent Site, Agent Time, SMS Assigned Sites
> - Package: Icon
> - Program: Icon

Configuring properties on the Criteria tab

You use the **Criteria** tab of the **Query Statement Properties** dialog box to specify the criteria by which the query selects records to display. Criteria are based on attributes of the object type, a relational operator, and a value.

The criteria for the example query statement described earlier, which returns all clients with Pentium III processors and with more than 1.5 GB of free disk space, is shown below as it appears on the **Criteria** tab in the Query Design view:

```
Processor.Name is like "%Pentium III%"
and
LogicalDisk.FreeSpace (MBytes) is greater than 1500
```

To create the criteria for the example query, perform the steps in the following procedures.

To select criterion type

1. In the **Query Statement Properties** dialog box, click the **Criteria** tab, and then click **New**. The **Criterion Properties** dialog box opens.
2. In the **Criterion type** list, click a criterion type. For the example query, click **Simple value**.

   The criterion type tells the processor what to expect for a criterion. For more information, see the "SMS Criterion Types and Values" section earlier in this chapter.

To select attribute class and attribute

1. In the Criterion Properties dialog box, click Select.
2. In the **Select Attribute** dialog box, click an attribute class in the **Attribute class** list. For the example query, click **Processor**.
3. Click an attribute in the **Attribute** list. For the example query, click **Name**.
4. Click **OK** to close the **Select Attribute** dialog box.

► To select a relational operator

1. In the **Criterion Properties** dialog box, click an operator in the **Operator** list. For the example query, click **is like**.
2. Click **OK** to close the **Criterion Properties** dialog box.

> **Note**
> There are four data types for SMS queries: numerical, date/time, string, and parameterized. Each data type has its own list of relational operators. Only the list of operators that applies to the selected attribute's data type is displayed. For more information, see the "SMS Relational Operators" section earlier in this chapter.

► To select a value to compare with the attribute

1. In the **Value** box, enter a value for the query to compare with the attribute that you have selected. For the example query, type **%Pentium III%**.

   –Or–

   Click **Values** to select from a list of available values.

   If a list of values exists for the attribute you chose, that list appears in the **Values** dialog box.
2. Click OK to close the **Criterion Properties** dialog box.

> **Note**
> The SMS Provider can run out of memory while caching a large result set. To avoid this, and to maintain performance, the Query Builder limits the number of values displayed in the **Values** dialog box to the first 2000. You can override this by changing registry settings. For more information, see article number 269201 in the Microsoft Knowledge Base at http://support.microsoft.com.

For more information about attribute classes, attributes, and values, see the "SMS Criterion Types and Values" section earlier in this chapter.

Create additional criteria

By completing the previous steps you have created the following expression, shown as it appears on the **Criteria** tab in the Query Design view:

```
Processor.Name is like "%Pentium III%"
```

Often, your query requires more than one criterion. In the previous example, the query returns all clients that have Pentium III processors. To modify the search to include those Pentium III processors that have 1.5 GB of free disk space, you must add another criterion. You can add as many criteria as you want, and each one further limits (AND, NOT) or expands (OR) the query. In the example, create a second criterion with the following properties, repeating the instructions in the previous steps if necessary:

- Criterion type of **Simple Value**

- Attribute class of **Logical Disk**
- Attribute of **Free Space**
- Operator of **is greater than**
- Value of **1500**

The second criterion appears below the first criterion as follows:

```
Processor.Name is like "%Pentium III%"
and
LogicalDisk.FreeSpace (MBytes) is greater than 1500
```

Choose the logical operator

By default, the AND operator connects the two criterion. In the **Query Statement Properties** dialog box, click the **And Or** button to replace the AND with OR. Select one of the expressions and click the **Not** button to insert NOT before the expression. For the example, leave the default AND as the logical operator.

Choose parentheses

In the example, there are no parts of the criteria expression that require grouping. Grouping with parentheses is used to clarify the meaning of expressions and to cause the expression or expressions within the parentheses to be evaluated first. If your query statement requires parentheses, highlight the expression or expressions that you want to place within the parentheses and click the **Parentheses** button.

By following these steps, you have created the following expression, shown as it appears on the **Criteria** tab in the Query Design view:

```
Processor.Name is like "%Pentium III%"
and
Logical Disk.FreeSpace (MBytes) is greater than 1500
```

To view the full query in the Query Language view, click **Show Query Language** in the **Query Statement Properties** dialog box.

To configure the query to return only clients running Windows 2000 Professional with Pentium III processors and that have greater than 1.5 GB of free disk space, you must limit the query to the **All Windows 2000 Professional Systems** collection.

► To limit the query to a collection

1. Click **OK** to close the **Query Statement Properties** dialog box and return to the **Query Properties** dialog box.
2. On the **General** tab, in the **Collection Limiting** area, click **Limit to a collection**.
3. Click **Browse**, and in the **Browse Collection** dialog box, click the **All Windows 2000 Professional Systems** collection.

Note

When you limit a query to a collection, the query is limited only to the collection you specify and is not limited by any subcollections of the specified collection.

For more information about limiting collections, see the SMS Help.

Creating Queries Against Multiple SMS Object Types

When you create a query by using the SMS Query Builder, you are limited to using the attributes of only one SMS object type at a time. You can use the **<unspecified>** object type to query against more than one SMS object type at a time.

When you use the **<unspecified>** object type, you can only create a query by using WQL in the **Query Language** view. You can use this to create free-form WQL queries to run against more than one SMS class. You must have a good understanding of WQL to use this feature.

► To create a WQL query against multiple SMS object types

1. In the SMS Administrator console, navigate to **Queries**.
2. Right-click **Queries**, point to **New**, and then click **Query**.

   The **Query Properties** dialog box opens.
3. In the **Object Type** list, click **<unspecified>**, and then click **Edit Query Statement**.

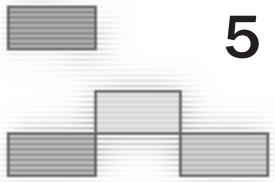
   The **Query Statement Properties** dialog box opens in the **Query Language** view.
4. In the **Query statement** box, type a valid WQL query statement.

The following is an example of a WQL query that queries both the **System Resource** and the **User Resource** SMS object types:

```
SELECT R.Name, U.UniqueUserName FROM SMS_R_System R, SMS_R_User U WHERE
R.LastLogonUserName=U.UserName
```

CHAPTER 5

Distributing Software



Chapter 3, "Understanding SMS Features," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* introduced the concepts behind Microsoft® Systems Management Server (SMS) 2003 software distribution, including:

- The general benefits of automating software distribution using SMS.
- The major components involved in SMS software distribution.
- The issues that software distribution can face, and that a proper deployment of SMS can minimize.

Software distribution consists of a series of specific but flexible tasks. This chapter describes those tasks, the preparations you must make to perform the tasks, and the procedures to distribute software.

In This Chapter

- Preparing to Distribute Packages
- Managing Packages
- Managing Advertisements
- Monitoring Software Distributions
- Using Software Distribution Tools and Wizards
- Running Advertised Programs on SMS Clients
- Software Distribution Common Practices
- Software Distribution Best Practices

Preparing to Distribute Packages

There are several tasks that you must perform before you distribute any packages in your SMS site. You must configure the Software Distribution Agent that runs on each SMS client in your SMS site. Similarly, you must configure the Software Distribution Component that runs on the SMS site server. There are also considerations for preparing SMS site systems.

This section includes the following tasks to perform before you distribute packages:

- Configuring the Software Distribution Agent
- Preparing client access points (CAPs), management points, and distribution points
- Preparing collections
- Preparing security
- Configuring the Software Distribution Component

Configuring the Software Distribution Agent

When software distribution is enabled, SMS installs the Advertised Programs Client Agent on all Legacy Client computers within the site, and enables the Advertised Programs Client Agent on all Advanced Client computers within the site. Before using SMS software distribution, examine the configuration of the Advertised Programs Client Agent and adjust the configuration if necessary. Within the **Properties** dialog box of the client agent, you enable or disable software distribution and set the interval for the client agent to check for newly advertised programs. You can also set up countdown and notification options when advertised programs are received and ready to run. Options that you select apply to all client computers in the site.

Enabling and Disabling Software Distribution

If you used SMS Express Setup, software distribution is enabled for the site. If you used SMS Custom Setup, software distribution is disabled. You can enable or disable software distribution at any time. The agents are not installed on the clients until the next client refresh cycle.

► To enable or disable software distribution

1. From the SMS Administrator console, navigate to **Client Agents** in the site settings for your site.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
    └ Site Settings
     └ Client Agents
```

2. Right-click **Advertised Programs Client Agent**, and then click **Properties**. In the Advertised Programs Client Agent **Properties** dialog box, use the **General** tab to perform these tasks:
   - To enable software distribution to clients, select the **Enable software distribution to clients** check box.
   - To disable software distribution to clients, clear the **Enable software distribution to clients** check box.

Setting Advertisement Options for SMS Clients

When you configure the Advertised Programs Client Agent, you can configure options that change the way your advertisements are displayed on client computers.

Set an interval for the client agent to check for new advertised programs

On the **General** tab, you can set intervals used by the Legacy Client and Advanced Client agents to check for newly advertised programs. The default interval is 60 minutes. Valid entries range from five minutes to one year.

Open Add or Remove Programs

On the **General** tab, you can specify that for Advanced Clients, the **New program notification** icon opens **Add or Remove Programs**. Advertised programs are always listed in both the Add or Remove Programs item in Control Panel and in Run Advertised Programs (on Advanced Clients) or the Advertised Programs Wizard (on Legacy Clients). When users are notified of new advertised programs using the new program notification icon in the notification area, they can double-click the icon to determine what advertised programs are available. For users on Advanced Clients, if this option is set, Add or Remove Programs is opened. If it is not set, Run Advertised Programs is opened. On Legacy Clients, the Advertised Programs Wizard is always opened. For more information, see the "Running Advertised Programs on SMS Clients" section later in this chapter.

Require that client computers use the settings you configure

On the **General** tab, you can specify whether users on Legacy Clients can override the software distribution client agent settings that you configure. Users on Advanced Clients must use the site-wide settings.

Display a visual indicator when new advertisements are received

On the **Notification** tab, you can specify that a dialog box appears when new advertisements are received. This applies to the Legacy Client only.

Play a sound when new advertisements are received

On the **Notification** tab, you can also enable an audio alert when new advertisements are received. Advanced Clients do not play sounds for any SMS events.

Provide a countdown when scheduled programs are set to run

On the **Notification** tab, you can enable a countdown dialog box when scheduled programs are about to run, and you can configure the countdown length. By default, the countdown runs for five minutes. Valid entries range from one to 60 minutes. The countdown starts at the time the advertisement is scheduled for, and the program runs when the user starts the program or when the countdown ends.

Play countdown sounds

On the **Notification** tab, you can set the system to play sounds during the countdown period. This setting applies to Legacy Clients only. Advanced Clients do not play sounds for any SMS events.

Show a status icon on the notification area for all system activity

On the **Notification** tab, you can set the notification area of the operating system taskbar to show a status icon when new advertisements are received.

For more information, see the "Running Advertised Programs on SMS Clients" section later in this chapter.

Preparing CAPs, Management Points, and Distribution Points

To ensure that a program can be advertised and run successfully, you must ensure that at least one client access point (CAP) or management point and at least one distribution point are available to the members of the target collection. As a preliminary task, examine the CAPs, management points, and distribution points in your SMS hierarchy, and consider adding or removing them as necessary. You accomplish this by:

- Preparing CAPs or management points.
- Preparing distribution points.
- Optionally, managing distribution point groups.

For information about creating new CAPs and configuring CAPs, see Chapter 15, "Deploying and Configuring SMS Sites," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

To add or change CAPs or distribution points, navigate to **Site Systems** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
    └ Site Settings
     └ Site Systems
```

Preparing CAPs and Management Points

Before distributing your package, examine all of the CAPs and management points in your SMS hierarchy. You can add or remove them if necessary. Prepare the CAPs and management points you want to use at the preliminary stage of the process, so they will be ready when you advertise a program.

At installation, SMS assigns the CAP role to the site server. You must create additional CAPs as required to provide access to all computers running the Legacy Client. You can reduce the load on the site server by creating additional CAPs, and by removing the CAP role from the site server.

Note

SMS 2003 does not automatically create management points when you install a site. You must create management points as required to provide access to all computers running the Advanced Client.

For information about creating SMS site systems, see Chapter 15, "Deploying and Configuring SMS Sites," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Preparing Distribution Points

Distribute your package, examine the distribution points in your SMS hierarchy, and add or remove them as necessary. Configure all of the distribution points that you want to use at the preliminary stage of the process so you can select from existing distribution points when you distribute packages.

At installation, SMS assigns the distribution point role to the site server. You can create additional distribution points to reduce the load on the site server and provide access to all client computers in your site. If software distribution in your SMS system includes multiple sites, specify a distribution point in each site to ensure access by client computers and to distribute the load. For more information about distribution points, see Chapter 15, "Deploying and Configuring SMS Sites," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

If you use the common SMS package shared folder on distribution points, when the first package is sent to a distribution point, the distribution point is given the share name *computername*\SMSPKG*driveletter*$ on the NTFS drive that contains the most available space.

Note

If there is not enough space on any distribution point drive to store the package, the software distribution process stops.

On this share, each package is stored in a separate folder that is identified by the package ID number. If the drive becomes full and another drive is available, SMS automatically creates an additional distribution point share on the available drive and puts the package there.

To make it easier to identify and organize related packages, you can instead have SMS store packages in a share distribution folder, whose name you specify. To control which drive either the default or custom package folder is created on, assign the distribution point role to a server share. For more information, see the "Set Package Properties" section later in this chapter, and the SMS Help.

Enabling Background Intelligent Transfer Service

By using Background Intelligent Transfer Service (BITS), Advanced Clients can transfer files from BITS-enabled distribution points and to any management point in an efficient and reliable manner. BITS is especially beneficial to software distribution, which often requires downloading large packages to clients. Those downloads can easily use all of the network capacity of a dial-up link for a long time. And the dial-up link might be disconnected in the middle of a package download. The full benefits of BITS are described in Chapter 4, "Understanding SMS Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

To enable BITS for software distribution, select the **Enable Background Intelligent Transfer Service (BITS)** option on the **Properties** dialog box for your distribution points if the distribution points need the software. For more information, see Chapter 15, "Deploying and Configuring SMS Sites," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Advanced Clients automatically use BITS if it is available. You can set an option on advertisements so that Advanced Clients will download the full package to a local cache before starting to run it. If the distribution point is not local but has BITS enabled, BITS is used to download the package. Downloading the package is a good option for a package large enough that the user will notice the effect. It is also good for a package that might not be downloaded during the time the user is connected to the network. For more information, see the "Running Advertised Programs on Advanced Clients" section later in this chapter.

Enabling Protected Distribution Points

Distribution points can be configured so that they are the distribution point used by clients within certain boundaries. Clients outside of those boundaries cannot use the distribution point.

To protect a distribution point in this way, select the **Enable as a protected distribution point** option in the **Properties** dialog box for your distribution point.

Managing Distribution Point Groups

Distribution point groups are a set of distribution points that you can manage as a single entity. Distribution point groups are helpful when the number of distribution points you usually work with is large enough to be inconvenient to work with individually. You can use distribution point groups to quickly create a diverse collection of distribution points, such as those in multiple sites.

Note

Distribution point groups are useful at the site the SMS Administrator console is connected to.

If you want to use a regular set of distribution points, you can create a group of all these distribution points, and then assign packages to the distribution point group, instead of to the individual distribution points.

Note

Distribution point groups cannot be used to remove distribution points from packages or to refresh packages on distribution points.

Before you distribute software, examine all of the distribution point groups at your site, and then add or remove distribution points if necessary. Configure all of the distribution point groups you want to use at the preliminary stage of the process, and then select from existing distribution point groups when you distribute software. You can create as many distribution point groups as you need.

For more information about distribution point groups, see Chapter 15, "Deploying and Configuring SMS Sites," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Preparing Collections

Before you distribute software, examine all of the collections in your SMS hierarchy and adjust them if necessary. Prepare the collections you want to use at the preliminary stage of the process so you can select from existing collections when you distribute software.

When you distribute a software package, you must identify the target collection of client computers, users, or groups that will receive the advertisement. Each advertisement specifies a single target collection, but you can also choose whether to distribute to subcollections of the target collection. A variety of commonly used collections is provided with SMS 2003. For optimal results, create collections that reflect how your organization organizes users, user groups, and computers for software distribution. After a collection is created, you can use it whenever it represents the appropriate target group for your package. When client computers are added, removed, or changed within sites, SMS evaluates the collections so that each collection is always current. The collection evaluations are performed on a schedule that you can modify. Changes in collections are automatically reflected in their corresponding advertisements.

You will probably maintain collections for groups of computers that perform similar work. Create collections that represent specific user groups or administrative groups if they are often used as criteria for software distribution. For more information about creating and working with collections, see Chapter 4, "Managing Collections and Queries."

Important

SMS 2.0 or SMS 1.2 16-bit clients that are identified by user accounts or user groups in your collections will not receive programs sent to them using the software distribution feature. Only 32-bit clients can receive software distribution programs based on user accounts and user groups.

Collections that contain query-based membership rules are evaluated at the site where they are created, and at any child sites to that site. For this reason, query-based collections are useful for guaranteeing that the advertised program is targeted to all computers that meet the criteria.

Note

Query-based collections are not appropriate for situations that require a greater degree of control. For example, if you have a limited number of licenses for a particular software application, you would not want to use query-based collections to distribute that software. Instead, you can use a collection with assigned resources for the advertisement target.

Choosing from Existing Collections

To choose a target collection from existing collections, navigate to **Collections** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Collections
```

Examine each collection and subcollection. If you find a collection that includes the complete list of client computers you want to target for the distribution, you do not have to create a new collection. Otherwise, create a new collection. For more information about creating or modifying a collection, see Chapter 4, "Managing Collections and Queries."

To examine the properties of any collection, right-click the collection and click **Properties**.

Note

To create a collection, you must have **Create** permission for collections. To advertise a program to a collection, you must have **Advertise** permission for collections. For more information, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Subcollections

The organization of collections and subcollections is similar to nested distribution lists in an e-mail program. Any collection can be made a subcollection of any other collection, because the query that creates the subcollection is entirely separate from the query that creates the collection.

When you create an advertisement that specifies a collection that has one or more subcollections, you can decide whether to distribute to the subcollections. For more information about subcollections, see Chapter 4, "Managing Collections and Queries."

To include subcollections in a software distribution, navigate to your advertisement in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Advertisements
```

Right-click the advertisement you want to modify and click **Properties**.

- To include members of subcollections in an advertisement, on the **General** tab, select **Include members of subcollections**.
- To exclude members of subcollections in an advertisement, on the **General** tab, clear **Include members of subcollections**.

Preparing Security

Before distributing software, ensure that administrators and users have sufficient rights to run the programs you advertise. For more information about permissions, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

SMS Administrator Console Security

With SMS 2003, you can specify security rights for working with collections, packages, and advertisements. You make these specifications from the SMS Administrator console. An SMS administrator's effective rights to work with an advertisement are determined by the rights the administrator's account has been granted for the collection, package, and advertisement security objects. This type of security model is called cumulative or additive.

Table 5.1 shows the minimum effective security rights that are required on the collection, the package, and the advertisement.

Table 5.1 Minimum Effective Security Rights for Software Distribution

To gain this effective advertisement right	You must have these rights		
	Collection right	Package right	Advertisement right
Read	Read	Read	Read
Modify	Advertise	Read	Modify
Delete	Read	Read	Delete
Create	Advertise	Read	Create or Administer

Package Access Accounts

SMS creates package source directories on distribution points with access permissions that, by default, make the package source files available to all users. Package access accounts are provided to restrict access to the files. You can grant a user or user group the permissions they must have to run the program. If you distribute software to a group of client computers, you must determine which users or user groups are likely to be logged on to each client computer.

Usually, you do not have to restrict access to the package source files, but if the files contain sensitive information, package access accounts can provide greater security. Also, if you must protect the files from sophisticated users who navigate to a distribution point and run programs that have not been advertised to them, use package access accounts. You can specify the following access levels to user groups or accounts that have permission to access to the package.

Table 5.2 Security Access Levels for Packages

Access level	Description
No Access	Prevents the account from reading, writing, or deleting files in the package folder on the distribution point.
Read	Enables the account to view and copy files, run programs, change directories within the shared folder, and read extended attributes of files. By default, SMS grants the generic Users account a **Read** permission to the package folder on the distribution point.
Change	Enables the account to change the contents and extended attributes of files and to delete files. **Change** permission is required for applications that write information back to the package folder on the distribution point.
Full Control	Enables the account to write the contents and extended attributes of files and to delete files. By default, the generic Administrators account has full control so that the SMS components can access the package folder on the distribution point.

By default, SMS creates generic Users package access accounts with **Read** access to the package shared folder on distribution points. If you specify your own package access accounts, ensure that all users who you intend to receive the advertisement are covered by the package access accounts you specify. Client computers without access to the package directories on distribution points will fail when attempting to run the advertisement.

As shown in Table 5.3, SMS creates the following generic package access accounts by default for each package.

Table 5.3 Package Access Accounts

Generic account	Rights
Users	Read
Administrators	Full Control

These generic package access accounts are mapped to operating system-specific accounts, and the appropriate rights on each operating system are applied to the package folder on the distribution point.

Table 5.4 Package Account Rights

Generic account	Operating system group
Users	Local Users
Administrators	Local Admins

Administrators can delete or modify these default access accounts. However, it is recommended that the Administrators account not be removed because it is required when SMS components update and modify the package. If you prefer not to use the generic package access accounts, you can set up your own accounts and specify one or more users or groups to be granted access to the package files on the distribution points. When the package is sent to distribution points, SMS will set security on the distribution point shared folder (...\SMSpkg*driveletter*$ by default).

To specify a package access account, navigate to **Access Accounts** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Packages
   └ package
    └ Access Accounts
```

Right-click **Access Accounts**, click **New**, and then click the kind of access account you want to create. You can create an operating system access account, or create a generic access account, which is mapped to an account on each of the systems, as described previously. The generic access account option is useful if you have deleted one or more of the generic access accounts. In the **Access Account Properties** dialog box, set the user or user group account that is allowed to access a package on the package's distribution points.

Important

If you remove a user from a group, it is necessary for the user to log off for the security changes to take effect. Otherwise, the user will still receive the advertisement.

To delete a package access account, navigate to **Access Accounts**, right-click the account you want to delete, and then click **Delete**.

Legacy Client Software Installation Account

When a user at a Legacy Client runs an advertised program locally, that program has the potential to run under two user contexts. Unless otherwise specified, the program runs under the logged-on user's context.

If this user account does not have sufficient privileges to install software on the client, configure the program to run with administrative credentials by using a local administrative account.

Note

This option can also fail in some cases, when the advertised program requires access to network resources other than the distribution point folder from which it is run.

Legacy Clients use the Legacy Client Software Installation account to support advertised programs on clients that require a special security context. Use this account when the advertised program meets the following criteria:

- The program must access network resources other than the distribution point from which it was run.
- The program is not an application coded to use SMS or other explicit connection mechanisms.
- The program requires administrative rights.

You must create the Legacy Client Software Installation account manually. Because this account is used to gain access to network resources required by the programs that are part of a package, you must:

- Create the account as a domain user account.
- Grant the account the rights needed to access the required network resources.

You can specify the Legacy Client Software Installation Account by navigating in the SMS Administrator console tree to **Site Settings**, pointing to **Component Configuration**, and then clicking **Software Distribution**. Then, for programs that require this account, configure the program by selecting its **Properties** dialog box, clicking the **Environment** tab, and then clicking **Use Software Installation Account**.

Advanced Client Network Access Account

The Advanced Client Network Access Account is a domain-level account that you can create for Advanced Clients. The Advanced Client uses this account when an advertised program needs to access a distribution point or a share on a server other than the distribution point. Consequently, this account must have the appropriate permissions on the share that the advertised program accesses. After the SMS client has tried using its computer account and the logged on user account to connect to the distribution point, the client attempts to connect using the Advanced Client Network Access Account.

You must create the Advanced Client Network Access account manually. Because this account is used to gain access to network resources required by the programs that are part of a package, you must:

- Create the account as a domain user account.
- Give the account the rights needed to access the required network resources.

You can specify the Advanced Client Network Access account by navigating from the SMS Administrator console tree to **Site Settings**, pointing to **Component Configuration**, and then clicking **Software Distribution**. Then, for programs that require this account, configure the program by selecting its **Properties** dialog box, clicking the **Environment** tab, and then clicking **Use Software Installation Account**.

Configuring the Software Distribution Component

Although the software distribution component is configured with defaults that are appropriate for most SMS installations, you can use the SMS Administrator console to specify:

- The drive on the site server where compressed package (.pkg) files created by SMS are stored. SMS compresses and stores packages that are distributed to other sites (and within sites if it is requested in the SMS Administrator console).
- The number of threads to allocate to package processing.
- The user name and password to use when your programs must be executed in a special security context.
- The retry settings for updating distribution points, CAPs, and management points.

To configure the SMS software distribution component

1. From the SMS Administrator console, navigate to **Component Configuration**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site name
    └ Site Settings
     └ Component Configuration
```

2. Right-click **Software Distribution** and select **Properties**.
3. Use the **Properties** dialog box to complete these configuration tasks:

Set a concurrent processing thread limit for the package

On the **General** tab, you can set a concurrent processing thread limit for the package. By default, the processing thread limit is three, but valid entries range from one through seven threads. As you allow more threads, SMS can process more packages concurrently.

For most installations, the default value is best. However, in cases where the site server's load and network bandwidth permit, you might want to increase the number of threads.

Note

Only one package will be compressed at a time, and only one will be decompressed at a time.

Set the compressed package storage location

On the **General** tab, you can set the compressed package storage location. This setting specifies where SMS stores compressed packages. SMS creates a compressed version of a package source folder when the package is sent to a different site, or when the package properties are set to create and reference a compressed copy of the package source folder. With this option, you can specify which drive on the site server SMS uses to store these compressed package files. For more information, see the "Package Compression" section earlier in this chapter.

Specify a Legacy Client software installation account

On the **General** tab, you can specify a Legacy Client Software Installation account. This option provides additional security and flexibility. You use the option by specifying an account that can run advertised programs on SMS clients on computers running Microsoft Windows® NT®, Windows 2000, Windows XP, or operating systems in the Windows Server™ 2003 family. By default, programs can run in the logged on user's context or in a local administrator account.

Specify an Advanced Client network access account

On the **General** tab, you can specify an Advanced Client Network Access Account. This option provides additional security and flexibility. You use the option by specifying an account that can be used to connect to distribution points. By default, distribution points are accessed using the logged on user's account if a user is logged on, or using the computer account if no user is logged on.

Set the number of retries for updating distribution points

On the **Retry Settings** tab, you can set the number of retries for the Distribution Manager to distribute package source files to distribution points. You set the number of retries and the delay intervals between them.

By default, retries are set to two, but valid entries range from one to 1,000 retries. The default retry delay value is 20 minutes, but valid entries range from one through 1,440 minutes. Change these settings to reflect the traffic on your network.

Note

Retries can generate significant network traffic. Generally, the lighter the network traffic, the more often you can set the number of retries.

Set the number of retries for updating CAPs and management points

On the **Retry Settings** tab, you can set the number of retries for the Advertisement Manager to distribute advertisements and package information to CAPs and management points. The available settings are the same as those for distributing package source files to distribution points.

Managing Packages

Every package consists of three tasks that you must create and manage: the package definition, the program that carries out the package tasks, and the process of distributing the packages to distribution points that are accessible by SMS clients that need to run the program that is targeted to them.

This section describes the following three tasks:

- Creating and managing packages
- Creating and managing programs
- Distributing packages

Creating and Managing Packages

SMS packages contain the files and commands you must use to run the programs in the package in addition to information such as which distribution points provide the package source files to client computers. For each package, specify:

- General information about the package, such as the name, version number, and vendor of the software.
- Whether the package includes package source files.
- If there are package source files, specify:
  - The package source folder that contains the package source files.
  - Whether SMS should create and store a compressed copy of the package source files.
  - How SMS stores the package source files on distribution points.
  - Whether and how often the package source files on distribution points must be updated.

► To create, modify, or delete a package

1. Navigate to **Packages** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Packages
```

2. Managing software distribution packages includes the following procedures:
   - Creating package source directories
   - Creating a new package
   - Creating a setup script

- Modifying an existing package
- Deleting a package

Note

To create a package, you must have **Create** or **Administer** permissions for **Packages.**

Create Package Source Directories

Programs use package source files when they run. In general, the programs that do not require package source files are programs that already exist on the client computers.

If a package contains source files and the site is running in Standard Security mode, you must create a package source folder that is accessible to the SMS Service account. Create this folder the same way you create any other folder on your computer. The package source folder can be a folder on a drive, or it can be the drive itself, including a CD drive. The package source folder can be on a remote computer, if the remote computer is accessible by the SMS Service account. For remote drives, always specify the package source folder by using the Universal Naming Convention (UNC).

If the site is running in Advanced Security mode, the source folder must be accessible from the site server using the site server's computer account.

When you have created a package source folder, you must designate it as such so that SMS will use it for package source files. For more information, see the "Set Package Properties" section later in this chapter.

Package Compression

SMS automatically compresses package source files when it sends the package to other SMS sites. By default, files distributed within the originating site are not compressed. When compressed packages are set to other sites, the other sites decompress the package, and then distribute it to the distribution points.

If the source files are on removable media such as CDs you can have SMS create a compressed version of the source files. SMS stores the compressed file and uses it instead of the original source files as a source for distribution.

To create a compressed version of the source files for your package, navigate to the package you want to compress from the SMS Administrator console. Right-click the package and select **Properties**. Click the **Data Source** tab and enter the source folder, if one has not already been specified. Then select **Use a compressed copy of the source directory**.

Caution

Changing the data source between using a compressed copy or the source folder for an existing package causes the package to be updated on the site's distribution points. Copies of the package at distribution points at child sites are not updated. If the files in the data source have changed in any way, the hash value used for the package will not match the hash value for copies that Advanced Clients download from those child sites. Those Advanced Clients will not be able to run the advertised programs that use the package. If you change the data source and the package files might have changed, and you must update all distribution points before changing the package data source.

Create a New Package

Software distribution requires a correctly formatted package definition. You can create a package definition by:

- Importing a package definition file using the Distribute Software Wizard or the Create Package from Definition Wizard.
- Using the **Package** properties page in the SMS Administrator console.

Import a Package Definition File

A package definition file is a specially formatted file describing a package and one or more programs. A package definition file is created outside the SMS Administrator console. Use a package definition file as an alternative to creating a package definition in the SMS Administrator console. If you already have a package definition file, import the file into a wizard. SMS immediately creates the package definition and programs. If you installed the **Package Automation Scripts** option when installing your SMS site, your site will include package definition files for commonly installed Microsoft applications with your SMS installation. Many Microsoft products and third-party applications ship with their own package definition files, and SMS Installer can create a package definition file for any packages it creates. Or, you can create your own package definition file by following the syntax rules and including the required entries as described in the package definition file topics included in the SMS Help.

Both the Distribute Software Wizard and Create Package from Definition Wizard can import package definition files for package creation. You can use a predefined package file by:

- Specifying the file when you create the package by navigating to **Packages** in the SMS Administrator console, right-clicking **New,** and clicking **Package From Definition**. In the Package from Definition Wizard you can select from package definitions that are included with SMS, or you can browse for a package definition file (.sms or .pdf files).
- Specifying a package definition file to be imported into the Distribute Software Wizard.

Import a Windows Installer Package

Windows Installer packages contain many of the details needed to create an SMS package. You can create an SMS package by importing a Windows Installer package in much the same way that you would import a package definition file, except that when browsing for package definitions, look for files with the extension .msi.

Set Package Properties

If you do not use a package definition file, you must create the package and set all the installation attributes through the SMS Administrator console. You can create a package by clicking **Packages** in the SMS Administrator console, pointing to **New**, and clicking **Package**. The **Packages** dialog box includes the following options:

Identification for the Package (name required)

Use the **General** tab to provide package details, including name, software version number, publisher, language, and comments. You can also change the icon associated with the package.

Specify the package source directory (required if there are package source files)

Use the **Data Source** tab to indicate that the package contains no package source files, or to specify the package source folder if package source files exist. You can use **Local drive on the site server** when package-related functions in the SMS Administrator console are always performed from the console the on site server. If the data source is a local drive on the site server, then the source folder cannot be changed, and programs cannot be added to packages from consoles that are not installed on the site server.

Caution

Do not specify a folder on a distribution point shared folder as a package source folder. This can cause an infinite loop of processing, resulting in excessive server load and possibly excessive network load. It will also cause the package source to be lost if the distribution point is removed.

You can also specify that the package be regularly updated on the distribution points.

Important

If you schedule weekly updates and you choose a day of the week, ensure that your start date matches the day of the week you choose. This helps ensure successful scheduling.

Specify the shared folder for package source files on the distribution point (optional, and applicable if there are package source files)

To specify whether to access the distribution folder through the common SMS package shared folder, or to specify your own shared folder name for this package, change the settings in the **Data Access** tab. When packages are stored in the common SMS package shared folder, each package is stored in a separate folder under this shared folder and is identified by its package ID number.

To make it easier to organize and track packages on distribution points, and to access the packages through means other than SMS, you can specify that SMS store a package in a shared distribution folder. Then you can create a hierarchy of directories to store related packages. For the shared folder name, you can assign either a shared folder that is unique among all packages, or a shared folder and a path, where the path must be unique among all packages.

Table 5.5 Examples of Shared Folder Names

Shared folder name\shared folder and path name	Resulting path on distribution point
Windows 2000	\\Dpservername\Windows 2000
Windows 2000\Windows 2000 Server SP3	\\Dpservername\Windows 2000\Windows 2000 Server SP3
Windows 2000\Windows 2000 Professional	\\Dpservername\Windows 2000\Windows 2000 Professional

To control which drive the default or custom package folder is created on, assign the distribution point role to a server shared folder instead of a server. For distribution points on server shared folder, if a shared folder name is entered for a package, it is treated as a path beneath the distribution point shared folder (\\MyServer\MyShare).

Table 5.6 Examples of Package Shared Folder Names for Windows 2000

Package shared folder name	Resulting path on distribution point
Windows 2000	\\MyServer\MyShare\Windows 2000
Windows 2000 Server SP3	\\MyServer\MyShare\Windows 2000\ Windows 2000 Server SP3

Note

Any shared folder name (or shared folder name and a path name) you create can be up to 64 characters, including backslashes (\).

Specify how to handle connected users at update time (optional)

On the **Data Access** tab, you can specify:

- Whether and how to disconnect all users from distribution points when package source files on those distribution points are updated. Not disconnecting users can lead to SMS not being able to update any distribution files that are open. However, disconnecting users can cause the user activities to fail.
- How many times SMS tries to update the package source files before disconnecting users.
- Whether to give users a grace period before they are disconnected.

Disconnecting users at update time ensures that advertised programs that have started running do not use a combination of files from the original version of the package and the updated version of the package, which could have unpredictable results. However, disconnecting users while an advertised program is running will cause that advertised program to fail.

The users that must be disconnected from the shared folder are sent a popup message warning them that they should stop using the distribution point. They are also notified when the update is completed so that they can resume using the distribution point. However, a user on the site server is not notified.

Note

Windows XP client computers do not get the notification of the disconnect.

Users on Advanced Clients that are downloading the advertised program to their download cache before implementation do not run a downloaded package that contains both original and updated files. If Advanced Client receives a new download SMS policy for the updated package, the current download of content is stopped, and a new download of content is started based on the new policy. If the Advanced Client does not receive a new download SMS policy, the download finishes but is rejected because a hash check will show that the downloaded package is not the same as the package that should have been downloaded.

Specify sending priority and preferred sender (optional)

When packages are distributed between sites, you must use senders. Senders are SMS thread components that use an existing connectivity system to communicate with other sites. Use this option to choose a sending priority and a preferred sender.

To set this option, use the **Distribution Settings** tab. For most installations, the default settings are best. However, if your package is very large or if a specific sender is faster or more convenient, designate a particular sender. For example, the Standard Sender handles large packages much more efficiently than a RAS sender does. For more information about senders, see Chapter 15, "Deploying and Configuring SMS Sites," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide.*

Set up Status Reporting (optional)

Use the **Reporting** tab to specify custom values used to match advertisements of programs from packages with their installation status Management Information Format files. Installation status Management Information Format files (MIFs) are generated by software distribution programs to supply information about the success or failure of their installation on 32-bit clients.

SMS clients, or programs distributed with SMS software distribution, typically generate installation status MIFs using the package details from the **General** tab. However, if the programs distributed with SMS software distribution create status MIFs that include name, version, or other values that do not match the values from the **General** tab, you must specify those values in the **Reporting** tab. If the installation status MIFs cannot be matched to values specified on the **General** or **Reporting** tab of any packages, the MIFs will be discarded, and you will not be able to determine the status of those advertisements.

Create a Setup Script

If you distribute a program that you want to run without user intervention, but the program typically requires user input, you must provide a setup script. To create a setup script, see the documentation for the software you are planning to distribute. With most professionally developed software, you can use command-line options, initialization files, transform files, or other techniques to control the installation of the software.

If these options are not available, then in many cases you can use SMS Installer or any other tools used to repackage software. For more information about SMS Installer, see Chapter 7, "Creating Software Installation Packages with SMS Installer."

Almost anything that can be done from the command line can be done with SMS software distribution. Conversely, for SMS to run a program, it must be possible to run the program from a command line. By determining the command-line options for the program, or by repackaging the program so that it can be run from the command line, you can also run it from SMS.

Ensure that all files required by the setup or scripting programs are included in the package source folder.

Any method used to automate a program's installation must be well tested in the variety of situations that can occur when the program is advertised to client computers.

Modify an Existing Package

Modifying the package definition will update the package definition at the site's child sites, but the package source files will not be updated. For more information about updating the package source files on child sites and distribution points, see the "Distributing Packages" section later in this chapter.

► To modify an existing package

1. From the SMS Administrator console, navigate to **Packages**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Packages
```

2. Right-click the package and click **Properties**. Use the package **Properties** dialog box to change the settings described in the "Set Package Properties" section earlier in this chapter.

Note

To modify a package, you must have **Modify** or **Administer** permissions for packages.

Delete a Package

When packages are no longer needed, delete the package to leave space for new packages. When you delete a package:

- All the programs within the package and all the advertisements for the package are also deleted.
- The package source files are deleted from the distribution points.
- Any compressed versions of the package source are deleted.
- Any package access accounts you have created specifically for the package are deleted.
- SMS security rights to the package are deleted.

After a package is deleted, new users or client computers joining the site will no longer receive notification or be able to run advertisements that reference programs in the package. If there is a chance that new users or client computers can use the advertisement and install the software, it makes sense to keep a package's programs advertised and on the distribution points until the programs are retired or replaced.

To delete a package

1. From the SMS Administrator console, navigate to **Packages**.

```
Systems Management Server
 └ Site Database
  └ Packages
```

2. Right-click the package you want to delete, and then click **Delete**.
3. Complete the Delete Package Wizard.

> **Note**
> To delete a package, you must have **Delete** or **Administer** permissions for packages.

When you remove a distribution point from the list, the distribution point's copy of the package source files is automatically deleted.

Creating and Managing Programs

Each software distribution package requires at least one program. Programs are commands that run on targeted client computers. After you create a package, you must create one or more programs. You can associate almost any activity with a program. For example, you can use a program to install new software on client computers, to run batch files, or to distribute data files. You can specify more than one program per package. For example, for the Excel package, you can create programs to perform a typical installation, a minimum installation, and a custom installation. Tasks associated with programs include:

- Creating a new program for a package

- Modifying an existing program for a package
- Deleting a program for a package

To perform any of these tasks, navigate to **Programs** under the package you want to associate with the program in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Packages
   └ package
    └ Programs
```

Create a New Program

To create a new program

1. Right-click **Programs,** click **New,** and then click **Program**.
2. Complete the following tasks in the **Program Properties** dialog box:

General tab

On the **General** tab, you can set any of the following options that apply to your package:

Identify the program (name required)

Name the program, and optionally, write a comment or select an icon for it.

Users can view the comment, so the comment can include any information relevant to users. For example, you might include a comment instructing users to call the help desk if they have any questions about the program.

You can use the program's icon to allow users to quickly find the advertisement in a list of available advertised programs. You can also define a convention to use certain icons for certain kinds of advertised programs (such as tasks, applications, system software, or other categories).

Command Line (required)

Specify the program's command line. This field can contain up to 255 characters. You can type in the command line or browse to the file you want to run.

When a program is run on a client computer, SMS first searches the package source files for the file in the program's command line. If the file is not found or if the package does not contain source files, SMS uses a defined set of search paths in order.

The command line can be a Windows Installer package, in which case Windows Installer runs the package.

The command line can also be any file name with a valid file extension. Any command line parameters in the command line are applied to the program that is used to run the file.

The command line does not:

- Use Dynamic Data Exchange (DDE).
- Apply security policy restrictions that would otherwise prevent files from being run using their file extension associations (such as .vbs).
- Use shell extension handlers.
- Open shortcut files or URLs.
- Run commands that are intrinsic to the operating system command prompt. For example, "copy" is not a valid SMS program command line. However, such commands can be included in a batch file, and the batch file can be used as the command line.

Start In (optional)

Specify a folder to start the program in. By default, the path of the distribution folder on the distribution point is added to the front of the folder path. You can also specify a full path or a fully qualified name of a remote folder. If an absolute path is specified, it must exist on or be accessible by every targeted client computer, or the program will fail.

Run (optional)

Set the mode in which the program is run. Choose **Normal**, **Minimized**, **Maximized**, or **Hidden**. By default, the program runs in **Normal** mode. **Normal**, **Minimized**, and **Maximized** are the display size. **Hidden** means that no window is displayed for the advertised program.

After running (optional)

Specify what happens after the program has completed. You can choose one of the following options:

- **No action required**—No restart or logoff occurs after the program executes. This is the default mode. (On 16-bit clients, this is the mode supported.)
- **SMS restarts computer**—After the program runs successfully, if a user is logged on, SMS prompts the user that the system must be restarted. If no user is logged on, SMS restarts the computer.

  If the program finishes and returns a Windows Installer return code of ERROR_SUCCESS_REBOOT_REQUIRED, the computer is restarted.

Caution

Unsaved data changes on the computer will be lost.

- **Program restarts computer**—The program restarts the client computer. The Advertised Programs Client Agent uses this option on client computers to enable the special status handling required when a program restarts itself.

- **SMS logs user off**—When the program finishes successfully, if a user is logged on, SMS prompts the user to log off. This option is useful if the program requires that users log off and then log on again before it can complete.

  If the program finishes and returns a Windows Installer return code of ERROR_SUCCESS_LOGOFF_REQUIRED, the user is logged off without notification.

Category (optional)

The user can find the advertised program in the "All Categories" and "What's New" categories, or an optional category that you specify. Advertised programs appear under the "What's New" category for up to 14 days, or until they are run.

Requirements tab

On the **Requirements** tab, you can set any of the following options that apply to your program:

Set Estimated Disk Space (optional)

You can set the estimated disk space. By default, it is set to **Unknown**. This value appears in the advertised program's properties on the clients, and helps the user decide if and when to run the advertised program. **Estimated disk space** is also used to calculate the estimated download time that is displayed to users when advertised programs are downloaded before being run.

Users cannot view the **Estimated disk space** if they select the advertised program in **Add or Remove Programs**.

Set Maximum Allowed Run Time

You can set the maximum allowed run time in minutes. By default, this value is set to **Unknown**. This value appears in the advertised program's properties on the client computer, and helps the user decide if and when to run the advertised program. If you leave the maximum allowed run time as unknown, SMS sets the actual maximum allowed run time as 12 hours.

If you set the **Maximum allowed run time**, SMS stops monitoring the advertised program if the program uses more than this amount of time on the client. This allows SMS to continue with other software distribution functions, such as running other advertised programs. SMS does not:

- Stop the program.
- Free up any drives that have been mapped for the advertised program.
- Free up any network connections made for the advertised program.
- Remove security rights granted to the SMS Client Token account, if any.
- Free up operating system resources used by SMS when running advertised programs.

If you do not set the **Maximum allowed run time**, SMS continues to monitor the program until it ends, or the computer reboots.

Users cannot view the **Maximum allowed run time** if they select the advertised program in **Add or Remove Programs**.

Specify Client Platforms Where Program Can Run (optional)

Select the setting to run the program without checking for any specific platform, or you can select a setting to specify platforms where the program can run.

Set Additional Requirements to Appear in Advertised Programs in Control Panel (optional)

Enter text that will appear in Advertised Programs in Control Panel with your advertisement. For example, you can tell users to shut down other applications before running this program. These requirements are not enforced.

Environment tab

On the **Environment** tab, you can set any of the following options that apply to your package:

Only when a user is logged on (optional)

Select this **Program can run** option to prevent the program from running if a user is not logged on. This is the default setting. This option is valid for client computers running Windows NT 4.0, Windows 2000, Windows XP, or operating systems in the Windows Server 2003 family. If the advertised program does not require administrative privileges (as set under **Run mode**), the advertised program is run in the user's context and the package is accessed on the distribution point by using the user's account.

Only when no user is logged on (optional)

Select this **Program can run** option to prevent the program from running until the user logs off the computer. This option is valid for client computers running Windows NT 4.0, Windows 2000, Windows XP, or operating systems in the Windows Server 2003 family. This option forces the program to run using the Client User Token account on Legacy Clients, or the local system account on Advanced Clients. If you have defined package access accounts, make sure the local Administrator or client network connection accounts can access the package folder on distribution points. If a user logs on while the installation is running, installation continues. The package is accessed on the distribution point using the SMS Client Connection Account on Legacy Clients, and the computer account on Advanced Clients.

Note

Programs that that are set to run when no user is logged on, but that are not assigned, are rejected as not valid by Advanced Clients and appropriate status messages are reported. Legacy Clients run these advertised programs when the user logs off.

Whether or not a user is logged on (optional)

Select this **Program can run** option to enable the program to run regardless of logged on user status. This option forces the program to run by using the Client User Token Account on Legacy Clients, or the local system account on Advanced Clients.

Run mode

Select whether the program will run with the logged on user's rights or with administrative credentials.

Run with administrative rights is automatically selected when **Program can run** is set to **Whether or not a user is logged on** or **Only when no user is logged on**. **Run with administrative rights** is optional if **Program can run** is set to **Only when a user is logged on**.

If **Run with administrative rights** is selected but **Use Software Installation Account** is not selected, then the program is run in the context of the Client User Token Account on Legacy Clients, or the local system account on Advanced Clients. The Client User Token Account is given administrative credentials for the program being run. For more information about security, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Important

If the advertised program is a Windows Installer package, the advertised program will fail on Windows NT 4.0 clients when the package is run with administrative credentials. SMS does not support running Windows Installer packages with administrative credentials on Windows NT 4.0 clients.

If **Program can run** is set to **Whether or not a user is logged on** or **Only when no user is logged on**, you can set the program to be run using the Software Installation Account. The Client User Token Account and local system account cannot access other computers. If your advertised program must access other computers, use the Software Installation Account. The distribution point is accessed using the SMS Client Connection Account on Legacy Clients or the computer account on Advanced Clients, so you do not have to use a Software Installation Account to connect to the distribution point.

If **Program can run** is set to **Only when no user is logged on** and **Run with administrative rights** is selected, you can specify whether the program requires user interaction with the program when it runs with the **Allow users to interact with this program (less secure)** option.

If you do not select **Allow users to interact with this program (less secure)**, the program runs in an administrative context and no user interface is displayed to the user. Leave this option unselected for all programs that do not display any user interface or that display a user interface but do not require the user to interact with the program.

If you select **Allow users to interact with this program (less secure)**, the user interface for the program is visible to the logged-on user and that user can interact with the program. Select this option only for programs that must run in an administrative context and that require the user to interact with the program.

It is strongly recommended that you use Windows Installer-based setup programs with per-user elevated privileges for installations that require administrative credentials but must be run in the context of a user that does not have administrative credentials. Using Windows Installer per-user elevated privileges provides for the most secure way of deploying applications with this requirement.

Important

If you advertise a program that is set to **Run with administrative rights** and you do not select **Allow users to interact with this program (less secure)**, the program might fail if it displays a user interface that requires a user to make a selection or click a button. In such a case, the user interface that the user is required to interact with is not visible to the user and can never be responded to. The program waits for user interaction until the program's **Maximum allowed run time** that is configured in the advertisement is exceeded. After the **Maximum allowed run time** is exceeded, the program's process is terminated on the client. If no **Maximum allowed run time** is specified, the program's process is terminated after 12 hours.

Note

During the period from when the program starts to run until the program's process is terminated, SMS will not start any other pending software distribution programs.

Set Drive Mode (optional)

Set the type of connection used for accessing distribution points. Options are **Runs with UNC name**, **Requires drive letter**, or **Requires specific drive letter**. Use the latter option if the program or your environment requires a specific drive letter.

Reconnect to distribution point at logon (optional)

Selecting this option causes the computer to reconnect the drive to the distribution point by using the specified drive mode each time the user logs on. This option is disabled by default. This option allows the program to complete installation steps, if required.

If the Advanced Client uses the Network Access account to establish the network connection, or if the advertised program is set to run with administrative credentials, the network connection will be remembered by the operating system when the user logs on, but the operating system will not be able to re-establish the connection. The operating system will display an error message indicating the network connection could not be re-established. You should not use this option if the Advanced Client uses the Network Access Account to establish the network connection, or if the advertised program is set to run with administrative credentials.

Advanced tab

On the **Advanced** tab, you can set any of the following options that apply to your program:

Run another program first (optional)

On the **Advanced** tab, select this option to indicate that this program requires another program to run. This option is useful to force dependencies, and for coordinating the installation of user and system-specific portions of an application's installation. Select the name of the desired package and program from the drop-down lists. This feature is not supported on 16-bit clients.

You can also specify that the other program is run every time the program being defined is run by setting **Run every time this dependent program runs**. This option is useful if the results of the other program must be updated every time the program being defined is run. For more information about running advertised programs with dependencies, see the "Program Dependency" and "Running Advertised Programs on SMS Clients" sections later in this chapter.

Note

If the program that you specify to run on a client computer fails, the dependent program will not run, and the Advertised Programs Client Agent generates an advertisement failure status message.

When the program is assigned to a computer (optional)

Select from these runtime preferences, which take effect when programs are assigned:

Run once for the computer (optional)

Selecting this option causes the program to run once on the computer. This is the default setting. This option applies to programs that are advertised to computers.

Run once for every user who logs on (optional)

Selecting this option causes the program to run once for each new user who logs on.

Suppress program notifications

The notification area icons and messages, and the countdown notifications, are not displayed for this program.

Disable this program on computers where it is advertised (optional)

If you select this option, SMS disables installation of the program on client computers. The program is still sent to distribution points, and it is still advertised, but it is not displayed as being available through any advertisements.

This is the preferred method for temporarily halting advertisements because it applies to all advertisements of the program and does not require client computers to refresh their list of advertised programs to take effect. When you disable this option, the program can run.

For more information about these options, see the SMS Help.

Windows Installer tab

You can use this tab to specify the Windows Installer product information to enable installation source management of this product. This selection dynamically updates SMS 2003 Advanced Clients Windows Installer network locations. This tab only applies to a per-product basis, and only updates source network locations for those Windows Installer products currently installed on the computer. It will support both per-computer and per-user installations. There are three primary methods by which the Windows Installer locations are updated:

- A distribution of an SMS program that contains Windows Installer information
- An administrator-defined recurring schedule
- An Advanced Client roams to a location supported by a different management point
- The subnet changes and more than 8 hours have elapsed since the last update

Maintaining a valid network source path for installed Windows Installer programs is valuable when the user needs to make an addition to their installed components. It is also valuable when a product repair is triggered, or when the original files are required as part of the patching process. This feature is not available for Legacy Clients. There is no interoperability with previous versions of SMS.

Modify an Existing Program

To modify an existing program, complete the following procedure.

To modify an existing program

1. Navigate to **Programs** in the SMS Administrator console and double-click the program you want to modify.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Packages
   └ package
    └ Programs
```

2. In the **Program Properties** dialog box, you can modify any of the fields described.

The changes are replicated to CAPs and management points immediately.

Delete a Program

Deleting a program also deletes all of the advertisements for that program. After you delete a program, new client computers entering the site will not receive notification of the program and cannot run the program. One of the advantages of SMS is that, without any administrator intervention, new client computers entering a site receive notification of all advertised programs for which they meet the collection criteria. This approach can save administrators time. In some cases, such as when new users must run the program, it makes more sense to keep a program advertised and on the distribution point until the program is retired or replaced.

To delete a program

1. Navigate to **Programs** in the SMS Administrator console, right-click the program you want to delete, and then click **Delete**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Packages
   └ package
    └ Programs
```

2. The Delete Program Wizard appears. You can use the wizard to make the decision if it is appropriate to delete your program.

Distributing Packages

To run an advertised program that uses source files, clients must have access to at least one distribution point for the package. You must specify at least one distribution point for each package you create that contains source files. Packages that do not use source files do not need distribution points set.

When you specify distribution points for a package, SMS places a copy of the package source files on each distribution point specified. SMS can also update package source files on distribution points according to your schedule, or you can update them manually.

If the target collection includes client computers that are members of different Windows domains in a site, either place the package on a distribution point in each domain, or set up a trust relationship between the domains at the site.

Caution

Do not place any files directly on the SMSPKGx$ shared folder, which is used by SMS. Files placed on the shared folder will be deleted when the package is deleted or moved. If you want to share folder files on a server that has a distribution point role, you must use a different shared folder.

SMS client software can use any distribution point at a site that the client computer can access.

Distributing packages to distribution points can require considerable network capacity, depending on the size of the package and network availability. Consider the timing of package distribution tasks and the number of distribution points to be updated at one time when doing package distribution tasks. SMS sender addresses can be used to control site-to-site network activity, but within the sites, the activities will occur as soon as possible.

The Manage Distribution Points Wizard

For assistance with distribution point management tasks, you can use the Manage Distribution Points Wizard. By using the Manage Distribution Points Wizard, you can:

- Copy the package to new distribution points.
- Refresh the package on selected distribution points.

- Update all distribution points with a new package source version.
- Remove the package from selected distribution points.

You can use the Manage Distribution Points Wizard to specify distribution points for your packages.

► To start the Manage Distribution Points Wizard

1. From the SMS Administrator console, navigate to **Distribution Points**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Packages
   └ package
    └ Distribution Points
```

2. Right-click **Distribution Points**, select **All Tasks**, and click **Manage Distribution Points**.

You can perform the following tasks with the Manage Distribution Points Wizard:

► Specify distribution points for a package and copy the package to the distribution points (required).

1. Select **Copy the package to new distribution points** and click **Next**.
2. The **Copy Package** screen displays all of the distribution points in the site and its child sites that do not currently have the package.
3. Select the distribution points or distribution point groups you want to use. When you complete the wizard, the process of copying the package to the selected distribution points begins. If you do not see the distribution points you want, you must create them as directed in the "Preparing Distribution Points" section earlier in this chapter.

Refresh the package on selected distribution points (optional)

Use this option if one or more distribution points become corrupted, or if you want to manually force copying the current package source version to a distribution point.

If a compressed copy of the package is kept at the originating site, that copy will be used for the package refresh. The package source will not be used. If a compressed copy of the package is not kept at the originating site, the package source will be used, but it will be presumed to be the same version of the files. The package version number will not be incremented. The package will not be redistributed to child sites. Instead, they will be refreshed from their local copies.

► To copy the current package source version to one or more distribution points

1. Select **Refresh the package on selected distribution points** and click **Next**.
2. The **Refresh Package** screen lists all of the distribution points that can be refreshed for this package. Then, select the distribution points you want to refresh.

Update all distribution points with a new package source version (optional)

Selecting this option increments the source version and source date displayed on the **Data Source** tab of the package properties.

When you first copy the package source file to the distribution point, it receives a version number of 1. Each time you update the files on the distribution point, the version number is incremented by 1.

If a compressed copy of the package is kept at the originating site, that compressed copy will be updated from the package source files.

If the package is assigned to distribution points in child sites, the new package source files will be compressed and sent to the child site for an update of the child site distribution points.

► To update all distribution points

1. From the SMS Administrator console, navigate to the Managing Distribution Points Wizard.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Packages
   └ package
    └ Distribution Points
```

2. Select **Update all distribution points with a new package source version** and click Next.
3. When you finish the wizard, the package at the distribution point is updated.

Remove a package from a distribution point (optional)

To remove a package from a distribution point, navigate to the Managing Distribution Points Wizard. Select **Remove the package from selected distribution points**, and then click **Next**. Select the distribution points you want to remove. When you finish the wizard, the process of removing the files from the distribution points begins.

If a package is removed from all distribution points for a child site, the package will also be removed from the site server. If a compressed copy of a package is kept at the originating site, and that package is removed from all distribution points, the compressed package will remain at the originating site server. For more information, see the "Delete a Package" section earlier in this chapter.

Delta Replication

When SMS 2003 updates the source files for a package, and the source files have already been distributed to child SMS 2003 sites, it sends the parts of the package that have changed since the last time the package was sent (originally, as an update, or as a refresh). Delta replication minimizes the network traffic between sites, especially when the package is large and the changes are relatively small.

Note

A file is considered to be changed if it has been renamed, moved, or its contents have changed.

Delta replication also occurs within each site to its distribution points. The files that have changed are transferred to the distribution points.

The originating site keeps the differences between the current version of a package and the previous five versions. If a child site has one of the previous five versions of the package, the originating site will send the appropriate changes to the child site. If the child site has an older version of the package, the originating site will send the entire package.

If the originating site sends the changed files for a package but the child site no longer has the package, or the package has been altered at the child site, the child site will send a status message to the originating site reporting the problem.

Note

If the SMS addresses to your child sites are closed when you are making changes to a package's source, do not update the distribution points multiple times before the time the addresses are opened. Each update will include the files from the previous update because the child sites will not yet have the previous update. The updates will include redundant files, wasting network bandwidth.

Managing Advertisements

After you create and distribute the package, you can advertise a program associated with that package to a target collection in your SMS site. This section describes the following tasks associated with managing advertisements:

- Creating advertisements
- Disabling or rerunning advertisements
- Ensuring package and advertisement integrity
- Maintaining packages and advertisements

Creating Advertisements

When you are ready to make a program in a package available to clients, you advertise the program to a target collection. In an advertisement, you specify:

- The package and program to run on the client.
- The target collection.
- The schedule for the program's advertisement to clients.
- When or whether the program is assigned.

SMS uses collections to determine which clients receive an advertisement for a program. Typically, you use a single collection many times as the target for many programs. If a client system or logged-on user is in the target collection, depending on the settings you specify in your advertisement, one of the following events occurs:

- SMS notifies the user that a program is available and takes no further action. The user can run the program immediately, schedule it to run later, or not run it at all.
- SMS notifies the user that a program is available. If the program has not been run by its scheduled time, SMS runs the program. The user can run the program immediately, schedule it to run before the assignment time, or do nothing and allow it to run at the scheduled time.
- SMS does not notify the user of the program and runs it at a scheduled time or after a specified event.

To run the program either as specified by a user or on an assigned schedule, the client's Advertised Program Manager components connect to one of the distribution points specified in the advertised package. For more information about collections, see the "Preparing Collections" section earlier in this chapter, and Chapter 4, "Managing Collections and Queries."

There are two ways to create an advertisement:

- Use the Distribute Software Wizard.

  This wizard guides you through the all the steps of performing a software distribution, including creating the advertisement.

- Create an advertisement.

  From the SMS Administrator console, you can create an advertisement by using any existing collection, package, and program.

To advertise programs

1. Navigate from the SMS Administrator console to **Advertisements**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Advertisements
```

2. Right-click **Advertisements**, and then click **Advertisement** from the **New** menu.
3. When the **Advertisement Properties** dialog box appears, complete it by performing these tasks:

Identify the Advertisement (required)

On the **General** tab, type a name for the advertisement. This is the name that users see.

Specify the software, what to do with it, and the target (required)

On the **General** tab, select the **Package**, **Program**, and **Collection**. If you have defined access accounts for the specified package, ensure that all members of the target collection have permissions through one of the package access accounts.

Set the Advertisement Start Time (optional)

On the **Schedule** tab, set the date and time the program will be advertised and made available to clients. By default, this option is set to the current date and time, and the program is available to run on the client immediately. When you coordinate this setting with the assignment information, you can set up different scenarios for running the program on the client. For more information, see the "Assigned Program Scenarios" section later in this chapter.

Set the Advertisement Expiration (optional)

To remove a program from the list of available programs after a specified period of time, click the **Schedule** tab, and then select **Advertisement will expire**. When a program expires, it is no longer run according to assignment schedules, and it no longer appears in the Advertised Programs Wizard, Advertised Programs Monitor, Run Advertised Programs, or Add or Remove Programs. The program is not deleted from the distribution points.

Note

If the expiration time is set to the past and the program has started running on the Advanced Client, scheduler does not send the expiration message. Content will be downloaded to the client, but the program will not run as expected. Ensure that expiration time is set to a time in the future.

Set the Advertisement Priority (optional)

On the **Schedule** tab, set the priority of an advertisement to control when it is sent to child sites. This priority is used with sender addresses to determine when the advertisement is sent to child sites.

Note

To advertise a program to clients, you must have these permissions:

- **Read security** access for the package that contains the program
- **Advertise security** access for the target collection
- **Administer** or **Create security** access for advertisements

For more information about the options used to advertise a program, see the SMS Help. For more information about processing at the client during software distribution, see the "Running Advertised Programs on SMS Clients" section later in this chapter.

Creating Advertisements with Assigned Programs

Assigning a program means that the program is mandatory, and it usually means that the program is run automatically at the client. You can base program assignments on a schedule, an event, or both. You can also set up a recurring assignment, so that the program is run every day at midnight, for example. When you configure advertisement-specific properties in the **Advertisement Properties** dialog box, additional options are available. Several of these options refer specifically to assigned programs:

Mandatory assignments (optional)

Advertised programs can be mandated to run on clients by giving them an assignment. Click the **New** icon to create an assignment.

Note

Advertised programs that are Windows Installer programs are listed in **Add or Remove Programs** in Control Panel. If these advertised programs have mandatory assignments, they will not display the **Remove** button in **Add or Remove Programs**. Users cannot remove mandatory Windows Installer programs.

Scheduled assignments

If you click **Schedule** when you create an assignment, you can use the **Schedule** dialog box to specify when the program is set to run. The start date and time can be in the client's time zone or in Coordinated Universal Time (UTC, formerly Greenwich Mean Time). You can also specify a recurring schedule if one is appropriate for your program.

Assign immediately after this event

Event-driven assignments are run when the specified event occurs. The following events are available:

As soon as possible

This option causes the assigned program to run after it reaches the client, and as soon as all required conditions are met. This event can occur immediately after the advertisement is received, for example, if the program is specified to run when no user is logged on, or after the current user logs off. The client has no control over this setting.

Assign on logon

The next time the currently logged on user logs on to the client, this setting causes the program to run automatically. The user has no control over this setting. For all users that are not currently logged on, the users must log on to receive the advertised program. After they log off and later log on again, the advertised program will run.

Assign on logoff

When the user logs off the client, this setting causes the program to run automatically. The user has no control over this setting. For all users that are not currently logged on, the users must log on to receive the advertised program, and then log off to run it.

Assignments are not mandatory over slow links

This setting suspends assignments for Legacy Clients on a slow link. By default, this check box is selected. Slow links are considered to be 40 Kbps or slower between the client and the distribution point.

Allow users to run the program independently of assignments

By default, advertisements with assignments are not visible to users. Selecting this option allows the assigned program to appear among the programs listed under **Advertised Programs**, **Run Advertised Programs**, or **Add or Remove Programs** in Control Panel. The user can run the program manually at any time before the time scheduled in the assignment. By default, this option is disabled.

Note

Unless this allow users to run the program independently of assignments option is selected, the assigned program is invisible to the user and is run without the user's control.

Most assigned programs are not displayed to users. Because users have no control over assigned programs, these programs usually do not appear in the Advertised Programs Wizard or the Advertised Programs Monitor. However, you can select the **Allow user to run the program independently of assignments** option. If you do, users can run the program voluntarily at any time until the program's scheduled run time. If the user does not run the program before the scheduled time, it runs without user intervention.

Assigned Program Scenarios

Assigned programs can be run in a number of contexts, and the properties of the programs determine which context is the most advantageous. Following are some of the scenarios for advertised programs, and how to set the properties for the most advantageous program installation.

Recurring Assignment

Some assigned programs must be run on a recurring schedule. An example of a recurring assignment is a virus scan program that is distributed and then assigned to run every night at midnight. In this case, you would create two programs within the virus scan package. Your first program would install the virus scan program, and the second program would run the virus scan program. The first program can run immediately or with any of the other options that reflect your site's requirements. Do not assign the second program as soon as possible. Instead, set a recurring schedule, such as every 24 hours at midnight. You could also create an additional program that would check for and install any updates to the virus scan program. Then you could assign the third program at an appropriate, recurring schedule.

Program Dependency

The scanning program can be made dependent on the installation program and advertise the virus-scanning program at the recurring interval you prefer. The first time the scanning program is scheduled to run, the dependency will cause the installation program to run. The scanning program will run as soon as the installation program stops running, and then on its recurring schedule.

Assignments Based on User Logon

Assignments can also work in conjunction with program properties. For example, you might want to upgrade every client at your site to a new service pack of Windows 2000, but minimize the disruption to users. In this case, within the properties of the service pack program, select the **Only when no user is logged on** option. Then, create an assignment to run the service pack program at the most convenient time for your organization. When the assignment time is reached, all systems with no user logged on will run the service pack program. All client computers with a logged on user will wait to run the program until the current user logs off. You can also choose to allow users to run the program manually before the program assignment time. To do so, select **Allow users to run the program independently of assignments** in the advertisement.

Event-driven Assignments and Scheduled Assignments

When an assignment is event-driven, such as one with a program that runs the **Only when no user is logged on** option, sometimes the conditions are not met at the scheduled time to run. For example, client computers that are turned off when an assignment time occurs, or that receive an advertisement after an assignment time has occurred, will run the program when all conditions for the program are met (for example, if a specific user logon state is required, when that user logon state occurs).

When an advertisement contains both scheduled and event-driven assignments, the resulting assignment is cumulative. For example, if you create a recurring assignment of once per day at 9:00 A.M. and also create an assignment at logon, the client will run the program the next time a logon occurs after 9:00 A.M, and at every subsequent logon.

Retrying Assigned Programs

If an assigned program fails on a client and the reason for the failure is something that might be corrected over time, SMS tries every ten minutes to run the assigned program. The Advanced Client retries for one week and the Legacy Client retries forever. A status message is sent to the site when the first retry is done, and another is sent when the advertised program eventually succeeds.

Advertisements to Advanced Clients

For advertisements to Advanced Clients, you have additional options on the **Advanced Client** tab in the **Advertisements Properties** dialog box:

Whether to run the advertised program from a distribution point or to download the package and then run it locally

By default, advertised programs are run from distribution points. If the client disconnects from the network the program will fail. If the distribution point supports Background Intelligent Transfer Service (BITS), setting the **Download before running** option ensures the package is downloaded to the computer before SMS attempts to run the advertised program. BITS resumes the download the next time the computer connects to the network, so disconnection from the network will not cause a problem. If the distribution point does not support BITS and the computer disconnects from the network, the download will fail.

Downloading the package before running it requires additional disk space on the clients. It can also take longer than running it from the distribution point if the advertised program requires a portion of the package's files.

Whether to use remote distribution points when local distribution points are not available

By default, advertised programs do not run unless a local distribution point is available. The client must be within the boundaries of an SMS site, and that site must have at least one distribution point with the package for the advertised program. The remote distribution points are at the client's assigned site.

You can allow the advertised program to run by setting the **Download from a remote distribution point before running** option. This is most appropriate when the package is large, or when the clients have slow network links to the remote distribution points.

You can also allow the advertised program to run from a remote distribution point by setting the **Run from a remote distribution point** option. This is most appropriate when the package is small, or the programs needed to run the advertised program are a small fraction of the package. For more information about downloading advertised programs, see the "Downloading advertised programs" section later in this chapter.

Disabling or Rerunning Advertisements

By right-clicking an advertisement in the SMS Administrator console, you can select a task to disable the program the advertisement is advertising. This option disables the program for all advertisements of the program, not just the currently selected advertisement

You can re-enable the program by right-clicking an advertisement with program that is disabled, and then selecting the task to enable the program.

Important

You can disable and re-enable a program at the site where the advertisement is created. Disabling or re-enabling a program at another site is not effective.

You can force an advertisement to be rerun by right-clicking an advertisement and selecting the task to rerun the advertisement. This will add an assignment to the advertisement to run the advertisement as soon as possible.

Note

You can rerun an advertisement if there are two or more assignments for a specific time.

You can do each of these tasks without using the task menu. Disabling and enabling a program is an option in the program's **Properties** dialog box. Adding an assignment is an option in any advertisement's **Properties** dialog box.

Note

When you click the **Advertisements** node in the SMS Administrator console, you will see a list of all advertisements. The last column indicates whether the advertisements are enabled or not.

Ensuring Package and Advertisement Integrity

After you create an advertisement, a package, and at least one program, ensure that the client can access and process the package. To do this, perform the following tasks.

Check the package content.

Ensure that the specified package source folder contains all of the files needed for all of the programs in the package to run. If the package supports more than one platform, ensure that the source folder contains all of the files needed to support all relevant platforms. Also, ensure that package source files include necessary batch programs or setup scripts.

Verify distribution point coverage.

If the package has source files, ensure that at least one distribution point is assigned to the package for each site in which the specified collection has members. Also, ensure that enough distribution points have been assigned to accommodate the load.

Consider restricting access to the distribution point.

If you want to restrict access to the package source on distribution points, do so by creating package access accounts. Specify the accounts broadly enough to cover all members of the collection. Then, either remove access from or delete the generic Users package access accounts.

Caution

Never delete the generic Administrators access account. It is used by SMS components to install and update the package on distribution points.

For more information, see the "Package Access Accounts" section earlier in this chapter.

Check server capacity.

Ensure that enough disk space is available on:

- The site server where the package is created.
- Any site servers that receive the package.
- Specified distribution points.

To check the capacity of the servers, you can check the free disk space in the **Site System Status** node of the SMS Administrator console, or you can run queries as described in Chapter 4, "Managing Collections and Queries."

Test the programs.

SMS cannot ensure that your programs will run after you distribute them. Before you finalize your software distribution:

- Test the programs by running them without SMS at a test computer.
- Test the distribution itself by creating a test package, and then having SMS copy the package to the distribution points. Create a test advertisement, and then run the program commands you previously tested on the test computer from a client.
- Run a sample distribution of the tested packages to a child site and run the program commands on a client of the child site.

Consider time zones and time settings.

If you advertise your software package to run at a predetermined time, then the program will run at that time within the client's time zone unless you set the package to run at UTC. When you create advertisements, consider the effect of time zones on your advertisement.

Also, be sure to synchronize the time settings on your clients with the time settings on your servers, especially if distributions are set to run immediately.

Maintaining Packages and Advertisements

The software distribution maintenance you perform depends on the nature of the distribution.

Periodic Updates

Some packages require periodic updates. For example, if you distributed a virus scan program to be run on a regular schedule, then as virus data files are updated, the package should be updated. In this case, if you have an assigned program for all your clients that runs each night at midnight, and if the source files are kept at the distribution point, then to update the package, you must update the source files at the distribution points. After you do, all of your clients will run the new virus scan the next time the application runs. If instead of distributing the files to the distribution points, you installed the files on each client, you must advertise a program that reinstalls the files. You do not have to change the advertisement that runs the virus scan. You must update the files on each client to have your clients run the new virus scan software on the same schedule.

Updates of Packages During Advertisements That Are Completed at Some Clients

The package that you are distributing might be an application that has an upgrade available, but which requires the original application to be installed. If not all of your users have installed the previous version, you can create a new package for the upgraded program that is dependent on the original program to run. If users have already installed the original application, the new package runs without a problem. If they have not installed the original program, the Advertised Programs Client Agent triggers the installation of the original program first.

Updates of Packages During Partially Completed Downloads

If a package is updated on a distribution point while clients are downloading it, the following occurs:

- The original download SMS policy for Advanced Clients is cancelled as soon as the new policy is received. If the client is downloading from a BITS-enabled distribution point, then the download is cancelled immediately.
- After the download is cancelled, if the Advanced Client has received an SMS policy for the updated package, it starts downloading the new package. If the Advanced Client has not received an SMS policy for the updated package, it tries to find a distribution point with the previous version of the package. If an Advanced Client finds such a distribution point, it restarts the download of the non-updated version of the package. If the Advanced client does not find a distribution point, it retries. The download is complete when a distribution point with the original package can be found or an updated download SMS policy is received and a distribution point with the updated package can be found, whichever occurs first.

If the package is refreshed on the distribution point instead of being updated, the behavior is the same, except that the Advanced Client is not required to receive an updated download SMS policy.

Package Removal

When all of your clients have installed the package, you might be able to safely remove the package from the distribution points. Before you remove a package, consider whether you should leave it on the distribution points for new clients or for clients that might require the package again (for example, for Windows Installer install-on-demand).

When you remove a package from all distribution points, the package still exists at the originating site. To delete the original package, use the Delete Package Wizard.

Although you might choose to keep a package at the originating site, you might want to delete one or more programs that exist in the package. To make this deletion, use the Delete Program Wizard. For more information about deleting packages, see the "Delete a Program" section earlier in this chapter.

Monitoring Software Distributions

After you distribute software, you can monitor the distribution by using the SMS status system. For example, if you advertise a program to run a virus scan each night at midnight, you might want to check every morning to see if all the clients have run the program. You can see this information at a glance in the main **Advertisement Status** console item. This console item displays every advertisement and includes status information.

The **Package Status** summary provides information about each package. You can select a package to see the information about a site-by-site basis. You can also select any site to see information for that package on a distribution point-by-distribution point basis. At any level (package, site, or distribution point), you can view the status messages that were used to create the statistics.

The **Advertisement Status** summary provides information about each advertisement, and then you can select an advertisement to see the information about a site-by-site basis. At either level (package or site), you can view the status messages that were used to create the statistics displayed in the status summary.

You might want to consider using SMS reports to monitor the status of packages and advertisements. SMS reports return a significant amount of useful status information. You can also use status message queries to directly obtain the status of advertisements or package distributions. You can use such queries in reports, to display status information in a more effective manner.

Note

You can determine which advertisements are targeted at an individual client by viewing the **Advertisements** tab in the client **Properties** dialog box of a client in a collection in the SMS Administrator console.

Using Status Summaries for Packages at Their Sites and Distribution Points

The Status System includes five console items describing the status of software distributions:

- Package status summary
- Advertisement status summary
- Package detailed information
- Advertisement detailed information
- Per-site package detailed information

In addition, you can view informational, warning, and error messages from each of these items.

The Package status summarizer level provides a quick view of how many distribution points have successfully made the package available, how many are still retrying, and how many have failed. If the numbers do not look right, you can double-click any package to see more information, or right-click and select **Show Messages** to see the informational, warning, and error messages that have been generated.

The **Package detailed information** console item provides site-by-site information for each site where the package was distributed. If you need more detailed information, you can double-click any site to see a distribution point-by-distribution point description. Or, you can right-click at any of these levels and select **Show Messages** to view the informational, warning, and error messages generated by the package at that level.

Monitoring Package Distribution

The SMS status system gives you a good view of how the distribution of your packages to distribution points is progressing. You can use status summaries for quick information and console items for more detailed information. Under each summary, you can get the information you need at the most appropriate level. SMS updates package status each time there is a change in the condition of a package.

► To check the package status

1. From the SMS Administrator console, navigate to **Package Status**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ System Status
   └ Package Status
```

2. To view the status messages associated with the package as a whole, select the package you want in the results pane, right-click, and select **Show Messages**. To view all of the status messages associated with that package, click **All**. To view selected messages, click **Errors**, **Warnings**, or **Info**.

3. To view package status information for a specific site, select the package you want in the console tree to display its information about a site-by-site basis. The package status information for each site appears in the details pane.
4. To view the status messages associated with a particular site for the package you selected, select the site you want in the details pane, right-click, and then select **Show Messages**. To view all the status messages associated with that site for that package, click **All**. To view selected messages, click **Errors**, **Warnings**, or **Info**.
5. To view package status information for a specific distribution point, select the package you want, and then select the site you want in the console tree. The package status information for each distribution point for the selected package and site appears in the details pane.
6. To view the status messages associated with a particular distribution point for the selected package, select the distribution point you want in the details pane, right-click the distribution point, and then select **Show Messages**. To view all the status messages associated with the distribution point for the package, click **All**. To view selected messages, click **Errors**, **Warnings**, or **Info**.

Monitoring Advertised Programs

You can simultaneously advertise multiple programs in multiple sites. All of the status messages generated by any component within your organization are collected by the status system, filtered, and processed to display meaningful information about each advertisement. You can either view the advertisement summary information, or you can view the status messages that produced the summary information.

► To check advertisement status

1. In the SMS Administrator console, navigate to **Advertisement Status**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ System Status
   └ Advertisement Status
```

2. To view advertisement status messages, in the details pane, select the advertisement you want, right-click it, and then select **Show Messages**. To view all the status messages associated with the advertisement, click **All**. To view selected messages, click **Received**, **Failures**, **Program Started**, **Program Errors**, or **Program Success**.
3. To view advertisement status information, select the advertisement you want in the SMS Administrator console tree. The advertisement status information appears in the details pane.

Advertised program success is divided into four columns: **Program Errors**, **Program Success**, **Program Errors (MIF)**, and **Program Success (MIF)**. If your advertised program generates status MIFs, you should use the **Program Errors (MIF)** and **Program Success (MIF)** columns. The advertised programs that generate status MIFs might also have results in the **Program Errors** and **Program Success** columns, but the **Program Errors (MIF)** and **Program Success (MIF)** columns are more accurate for advertised programs that generate status MIFs.

Important

Status for advertised programs that generate status MIFs that are run at SMS 2.0 clients reporting to SMS 2.0 sites appears in the **Program Errors** and **Program Success** columns. If the advertised programs generate both normal status and status MIFs, the status might include duplicate records for those clients.

Using Status MIFs

To provide additional status reporting, you can direct your advertised programs to generate status MIFs. SMS Installer has this option built in. The Windows 2000, Windows XP, and similar upgrade programs automatically generate status MIFs. You can add lines to your setup scripts to call Ismif32.dll, as described in the *Microsoft Systems Management Server 2003 Software Development Kit*. Or, you can use the Ismif32.exe program from the SMS Support Tools. For more information, see the relevant documentation for each of these options.

You might want to use status MIFs for several reasons:

- Default advertisement status reporting returns one of two possible values for each client: success or failure. For large or complex packages, you might want information specifically why an advertisement failed.
- The advertised program might return a status code that indicates success or failure. To distinguish between actual success and failure, you might have to incorporate additional logic into the package to verify success, and then create a status MIF that accurately reflects that condition.
- If the package requires a restart before the installation can complete, you might want a status message before the restart, in addition to after the completion of the advertisement. This way, you can identify computers that are stuck in the middle of the installation of the advertisement. You can use such additional status reporting to know what type of intervention is required to correct any computers with failed advertised programs.

Ismif32.dll is installed on every SMS 2003 client that has software distribution enabled, so it can always be used to create status MIFs. The following example demonstrates how to create a status MIF from a Windows Installer script using Ismif32.dll:

```
item: Call DLL Function
  Pathname=%WIN%\ismif32.dll
  Function Name=InstallStatusMIF
  Argument List=41filename
  Argument List=41publisher
  Argument List=41product
  Argument List=41version
  Argument List=41language
  Argument List=41serialnumber
```

(continued)

(continued)

```
  Argument List=41The install failed for no good reason!
  Argument List=010
  Return Variable=0
  Flags=00100000
end
```

When viewing advertisement status in the SMS Administrator console, you will find that the messages have different identifier codes and description strings if they are based on a status MIF rather than SMS's default advertisement status reporting. Status messages 10009 (success) and 10007 (failure) are based on status MIFs, and will have the additional information included with the status MIFs. Status messages 10008 and 10006 are the default advertisement status messages for success and failure, respectively.

The status MIFs generated on the clients must be saved in either the system %temp% or %Windir% directories. %Windir% is used if the user has sufficient privileges to write to that folder; otherwise the files are placed in the %temp% folder. The preprogrammed status MIF generation tools will automatically place status MIFs in these directories. If you generate status MIFs by using other techniques, you must ensure the status MIFs are placed in these directories.

The SMS client confirms that the status MIF it finds is meant for the advertised program that has just run by comparing the details in the status MIF with the details of the program's package, such as name and version. By default, SMS uses the details set on the **General** tab of the package's **Properties** dialog box. Not all possible values have to be specified in the status MIF, but any values specified must be exactly matched by the values in the package's **Properties** dialog box.

For SMS to collect two status messages for an advertised program, the **After running** option in the program's **Properties** dialog box must be set to **Program restarts computer**.

Status MIFs must have a file creation date after the advertised program starts running on the computer. Status MIFs cannot be created before running an advertised program. If multiple status MIFs are available, SMS will use the most recent one.

Using Software Distribution Tools and Wizards

SMS includes the following software distribution tools and wizards.

SMS Installer

You can use SMS Installer to create an executable file that you can add to a package and advertise to clients. SMS Installer creates a self-extracting file or Windows Installer file that includes the data and files for the software application and the installation script that you created using SMS Installer. By using the SMS Installer Script Editor, you can modify the installation script that SMS Installer creates.

SMS Installer does not create the package, distribution points, or advertisements within SMS, so you must use another method to perform these tasks. SMS Installer creates a package definition file that can be imported into SMS with either the Distribute Software Wizard or the Create Package from Definition Wizard. For more information about SMS Installer, see Chapter 7, "Creating Software Installation Packages with SMS Installer."

Distribute Software Wizard

The Distribute Software Wizard automates the complete software distribution process. With this wizard, you can accomplish all the steps needed to distribute software. You can also use this wizard to perform the following individual software distribution-related tasks:

- Create a package and program manually.
- Create a package and program from an existing package definition.
- Specify package source file options.
- Specify distribution points for the package.
- Select an existing target collection.
- Create a new collection.
- Add a resource to a new or existing collection of resources.
- Create an advertisement.

Each of these tasks might not apply to all software distributions.

To open the Distribute Software Wizard, navigate to it by right-clicking **Systems Management Server**, or any collection, resource, package, or program in the SMS Administrator console. Right-click the item you chose in the SMS Administrator console, select **All Tasks**, and then click **Distribute Software**.

The panes that appear depend on how you started the wizard. For example, if you start the Distribute Software Wizard by selecting a package from **Packages** in the SMS Administrator console, the wizard is set to use the selected package.

When the Distribute Software Wizard creates an advertisement, it sets the advertisement to not run when no local distribution point is available. If you want the advertised program to be downloaded before running, or to run from a remote distribution point, you must modify the advertisement after using the wizard.

The Distribute Software Wizard requires appropriate security rights. For more information, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Create Package from Definition Wizard

This tool uses a package definition file to create a package. You can use the package definition files included in SMS, create one by using SMS Installer, or create a package definition file yourself. For more information about package definition files, see the "Import a Package Definition File" section earlier in this chapter.

Manage Distribution Points Wizard

For information about this wizard, see the "Distributing Packages" section earlier in this chapter.

Advertised Programs Wizard

For information about this wizard, see the "Running Advertised Programs on SMS Clients" section later in this chapter.

Advertised Programs Monitor

For information about this Control Panel item, see the "Running Advertised Programs on SMS Clients" section.

Run Advertised Programs

For information about this Control Panel item, see the "Running Advertised Programs on SMS Clients" section.

Program Download Monitor

For information about this Control Panel item, see the "Running Advertised Programs on SMS Clients" section.

Add or Remove Programs

For information about this Control Panel item, see the "Running Advertised Programs on SMS Clients" section and the operating system Help.

Delete Package Wizard

For information about this wizard, see the "Delete a Package" section earlier in this chapter.

Delete Program Wizard

For information about this wizard, see the "Delete a Program" section earlier in this chapter.

Delete Collections Wizard

For information about this wizard, see Chapter 4, "Managing Collections and Queries."

Running Advertised Programs on SMS Clients

When the SMS policy for an advertised program becomes available on a management point used by targeted Advanced Clients, and those clients can also find the relevant package on a distribution point, the Advanced Clients will assess whether they should run the program and then proceed to do so, if appropriate.

Similarly, when an advertisement becomes available on a CAP used by targeted Legacy Clients, and those clients can also find the relevant package on a distribution point, then the Legacy Clients will assess whether they should run the program and then proceed to do so, if appropriate.

Running Advertised Programs on Either Client

The following elements are the same when running advertised programs on either Legacy Client or Advanced Client:

- Assessment of the advertisement and program to determine if they are currently relevant to each client
- Running advertised programs that are installation-based
- Running assigned advertised programs
- Running advertised programs that run when a user is not logged on
- The notification area interface
- Categories

Assessment of the advertisement and program to determine if they are currently relevant to each client

Advertisements are assessed by the clients to determine whether they are enabled, active, and not expired. Programs are assessed to determine whether they are enabled, active, and relevant to the operating system or service pack being run on the client. These assessments are performed whenever the client reevaluates advertised programs, which by default is once per hour.

Running advertised programs that are installation-based

Installation-based programs are always run through **Add or Remove Programs** in Control Panel. Programs are designated as being installation-based by setting **Display in Add or Remove Programs** on the **General** tab of the **Programs Properties** dialog box.

After an advertised program has been successfully installed from **Add or Remove Programs**, attempting to re-run the advertised program from **Add or Remove Programs** does not cause the program to reinstall.

Running assigned advertised programs

Assigned programs are initiated without user intervention.

The notification area interface

Both Advanced Client and Legacy Client use the notification area interface to notify the user of advertised programs.

Categories

Both Legacy Client and Advanced Client can use Categories. All advertised programs will appear in the **All Programs** category. Any advertised programs that have been advertised in the last 14 days will also appear in the **What's New** category.

Running Advertised Programs on Advanced Clients

Running advertised programs on Advanced Clients is different from running them on Legacy Clients in the following ways:

- Using the **Run Advertised Programs** item in Control Panel for non-assigned advertised programs.
- Checking the status of advertised programs that must be downloaded before being run by using the **Program Download Monitor** item in Control Panel.
- Configuring the software distribution agent on the client.
- Viewing properties of advertised programs.
- Running dependent programs.
- Using BITS and client-side caching by some advertised programs.
- Downloading advertised programs before they are run.
- Managing the download cache.

Run advertised programs

If the advertised program is set to do so, users are notified of new advertised programs by a notification in the notification area. If an advertisement for a program becomes available for a program that was previously advertised to the client and run successfully, the user is not notified in the notification area.

Advertised programs are always available in both the **Add or Remove Programs** and the **Run Advertised Programs** items in Control Panel.

Program download monitor

You can use the Program Download Monitor to perform the following tasks:

- Monitor package downloads for advertised programs.
- Cancel downloads.
- Set an advertised program with a package that is being downloaded to start automatically when the download is complete.

To run the Program Download Monitor, click the **Program Download Monitor** icon in Control Panel.

The Program Download Monitor displays a list of active downloads on the client.

Configuring the software distribution agents on advanced clients

The software distribution agent configuration cannot be changed through SMS-provided user programs on Advanced Clients. The Advanced Client uses the site-wide software distribution client agent settings unless specially overridden by an administrator.

For information about how to specially configure software distribution agent settings on Advanced Clients using administrator options, see Chapter 4, "Understanding SMS Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

The download cache can be managed on Advanced Clients by using the **Systems Management** item in Control Panel, if the user has administrative credentials on the computer.

Viewing properties of advertised programs

To view the properties of an advertised program, the user at the client can select the program in **Run Advertised Programs** and click **Properties**.

Users can also see advertised program properties from the notification dialog box when the advertised program is ready to run.

Program dependencies

You can set advertised programs to run another program first. If the other program has already run, the advertised program proceeds immediately.

Note

If you delete a program dependency, the parent program and advertisement are disabled.

If any of the programs require packages to be downloaded, the package download message is displayed to the end user (if appropriate) and the packages are listed together. The Program Download Monitor also lists all the packages to be downloaded. The cache must have sufficient space for all the packages. The program that is lowest in the dependency chain is downloaded and run, and then the next program in the chain is downloaded and run.

If any of the programs in the list of dependent programs does not run successfully, the sequence of programs after that program is stopped. The programs can be retried at any time.

BITS might be used by some advertised programs

When you specify properties for an advertisement, you can set an option to download the package before running it. This can be set for packages that are to be downloaded from local distribution points, remote distribution points, or both. If the package is downloaded, it is stored in the Advanced Client download cache. If the package is downloaded from a remote distribution point, and that remote distribution point is BITS-enabled, then BITS is used to transfer the package to the client. If the package is downloaded from a local distribution point, or the remote distribution point is not BITS-enabled, then SMB checkpoint/restart file copy is used.

If the package is not downloaded before running an advertised program, then the program is run directly from the distribution point. If the network link fails or is closed before the program has completed running, the advertised program will be unsuccessful. The SMS status system will record the failure and report it to the SMS hierarchy the next time the client connects to the network.

Downloading advertised programs

When an advertisement is created, it can be set so that the package for the advertisement is downloaded to Advanced Clients before the advertised program being run. The download can be set to occur depending on whether a local distribution point is available or not. A local distribution point is a distribution point for a site that the Advanced Client is currently in a local roaming boundary of. For more information about how clients find distribution points, see Chapter 4, "Understanding SMS Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

If the end user initiates the download, the user is shown a progress message that the user can hide. The progress message indicates how long the download will take. The length of time is an estimate that for the first 30 seconds is based on a 28.8 Kbps link. After the first 30 seconds, the estimate is based on the rate that the package is actually being downloaded.

Advertised programs can be targeted at computers or users. If a download starts for an advertised program targeted at the client computer, the download continues if the user logs off, and will continue if another user logs on. However, if a download starts for an advertised program targeted at the user, the download stops when the user logs off and does not resume until the original user logs back on.

If the advertised program is also advertised to another user that logs on, the download starts from the beginning for that user. The download for the original user continues from the point it left off when that user logs back on.

Downloads also stop when:

- The computer is stopped, or set into a hibernate or suspend condition.
- The network link drops.
- The package is removed from the distribution point.

Downloads resume automatically when the computer is started up again and a network link can be established to a distribution point with the package. If a download is started but then interrupted, the download must resume within seven days or the download is automatically cancelled.

If an advertised program expires or is disabled while being downloaded, the download finishes, but the advertised program is not run.

It is possible that an advertised program's package will be downloaded, the advertised program will start to run, and then a new download SMS policy will arrive at the client indicating that an updated package is now available. In this case, the advertised program will continue to run.

When a download is finished without using the BITS protocol, and the download is resumed, it starts at the beginning of the file that was being downloaded at the time the download was interrupted. This is also true if the download resumes from a different distribution point, even if the different distribution point uses BITS. For this reason, packages should not be based on a small number of large files, if possible. In the case of an SMS Installer or Windows Installer package, the instructions can be kept in a separate file and the source files in the package should be kept separately, instead of being included in the SMS Installer or Windows Installer file. If the software is provided in large files, then investigate whether the software has an administrative installation or similar option that allows expanding the large files into a folder tree with many separate files. The SMS package will then use that expanded version of the software as the package source.

For more information about checkpoint restart while downloading packages, see Chapter 4, "Understanding SMS Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Managing the advanced client download cache

Managing the Advanced Client download cache is important if the client downloads and runs new advertised programs, but the cache is too full of active downloaded packages.

When a package is downloaded it is placed in the cache and locked. SMS does not delete a package from cache if it is locked. A package is unlocked when either of the following events occurs:

- 30 days have passed and the program has not been run
- 24 hours have passed since the program was run

After SMS unlocks the package, it cannot be locked again unless it is discarded and then downloaded again.

When a package must be downloaded but the cache cannot accommodate the package, SMS checks the other packages in cache to determine whether deleting any or all of the oldest packages will free enough space to place the new package into the cache. If deleting any or all of the oldest packages does not free enough space, the new package is not placed into the cache. This might be the case if there is a package that is currently locked. If deleting any or all of the oldest packages does free enough space in the cache, SMS does so, and places the new package into the cache.

Users with administrative credentials on the computers they are using can manage the download cache. Users can change the size or location of the cache, or delete all current contents. These options are in the **Temporary Program Download Folder** section of the **Advanced** tab of the **Systems Management** item in Control Panel.

The download cache can also be managed with scripts. For more details about scripting client operations, see Appendix C, "Scripting SMS Operations," and the SMS 2003 SDK.

You can avoid managing the download cache on clients by:

- Setting the cache size to be sufficiently large for the packages that will be downloaded.
- Scheduling downloads so that they do not occur too frequently.

- Not using the download option for packages that can be run directly from the distribution points.

Running Advertised Programs on Legacy Clients

Running advertised programs on the Legacy Client is different from the Advanced Client in the following ways:

- The Advertised Programs Wizard is used for non-assigned advertised programs.
- The Advertised Programs Monitor is used for advertised programs after they have been run, started to run, or scheduled to run.
- Configuring the software distribution agent.
- Running dependent programs.
- Viewing properties of advertised programs.
- Scheduling when an advertised program is run.

Run advertised programs

If the advertised program is set to do so, users are notified of new advertised programs in the notification area. If an advertisement for a program becomes available for a program that was previously advertised to the client and run successfully, the user is again notified in the notification area.

Advertised programs are available in both the **Add or Remove Programs** and the **Run Advertised Programs** items in Control Panel except that Microsoft Windows 98 and Windows NT 4.0 clients do not display advertised programs in **Add or Remove Programs**.

Advertised Programs Wizard

When an advertised program is available on a Legacy Client, an **Advertised Programs** icon with the label **New Advertised Program(s) are available** appears in the client's taskbar notification area.

When a new advertised program is available at the client, the user can use the Advertised Programs Wizard to run the program immediately, or to reschedule the program. To start the Advertised Programs Wizard, the user at the client can do one of the following:

- Double-click the **New advertised programs available** icon in the notification area.
- Right-click the icon and select **Run Advertised Program Wizard** from the pop-up menu.
- In Control Panel, double-click **Advertised Programs**.

When a new advertised program is available, the **New advertised programs available** icon appears in the user taskbar notification area. When an advertised program runs on the client, the **Advertised program running** icon appears in the user taskbar notification area. When an advertised program counts down to run on the client, the **Advertised program about to run** icon appears in the notification area.

Advertised Programs Monitor

The Advertised Programs Monitor helps users perform the following tasks:

- Monitor program run status.
- Change configuration options for the Advertised Programs Wizard.
- View advertised program properties.

To run the Advertised Programs Monitor, the user can perform one of the following at the client:

- Double-click either the **Advertised program about to run** icon or **Advertised program running** icon in the notification area.
- Right-click the icon in the notification area, and then click **Open Advertised Program Monitor** from the pop-up menu.
- Click the **Advertised Programs Monitor** icon in Control Panel.

The Advertised Programs Monitor displays a list of all scheduled programs, all programs that are currently running, and all programs that have already run at the client. The run status of each program appears in the **Scheduled to Run** and **Last Run** columns.

Configuring the software distribution agents on Legacy Clients

When you configure the Advertised Programs Client Agent properties in the SMS Administrator console, you can specify whether users at clients can override the default settings. If you enable users to change the agent settings, users can specify:

- How often the client checks for new advertised programs.
- Whether to be notified visually or with an audible prompt when a new advertised program is available.
- When a scheduled program is about to run, whether a notification message appears, and how long before runtime to display it.
- Whether and when to play sounds for countdown notifications.
- Whether to show the status icon on the taskbar for all system activities.

The user can change the Advertised Programs Client Agent settings by selecting **System** from the **Advertised Programs Monitor** menu, and then clicking **Options**.

Viewing properties of advertised programs

To view the properties of an advertised program, the user at the client must do one of the following:

- Select the program in the **Advertised Programs Monitor**, right-click the program, and then select **Properties**.
- Select the program in the **Advertised Programs Monitor.** On the menu bar, select **Program**, and then click **Properties**.
- Select the program in the **Advertised Programs Wizard** and click **Properties**.

Users can also see advertised program properties from the notification dialog box when the advertised program is ready to run.

Program dependencies

Advertised programs can be set to run another program first. If the other program has already run, then the advertised program proceeds immediately. Otherwise, the other program is automatically run. The exception is if the other program requires that another program be run first, in which case this other program will be run first.

If any of the programs in the list of dependent programs does not run successfully, the sequence of programs after that program is stopped. The programs can be retried at any time.

The user can schedule when an advertised program will be run

After the advertised program has been scheduled to run, the user can see the advertised program in the Advertised Programs Monitor. The user can cancel the scheduled running of the advertised program by selecting it and then clicking **Unschedule** on the **Programs** menu.

Software Distribution Common Practices

Some common software distribution tasks with SMS:

- Distributing packages to a single user or computer
- Stopping an advertisement in an emergency
- Re-running an advertisement
- Running an advertised program on a regular basis on clients
- Using Windows Installer-based applications with SMS
- Running an advertised program in the user context but with administrative credentials
- Running an advertised program within a time window
- Running an advertised program without any user intervention
- Estimating how long a package transfer will take
- Expanding the target of advertisements

Distributing packages to a single user or computer

Sometimes it is necessary to distribute a package to a single computer. For example, this might be useful if a user is having problems with an application and reinstalling the application will help. A solution to this problem is to create a new collection that contains the user or a specific computer, and then create an advertisement of a program for the relevant package for that collection. However, that is somewhat time consuming and can result in the proliferation of many collections.

A better approach is to create a permanent collection and advertisement for the purpose of reinstalling the application. Then when a user requests a package reinstallation, you have to add the user or a specific computer to the collection. You do not have to create a collection or advertisement. The users or computers already in the collection will not receive the package again, because they received it when they requested it. Only the user just added to the collection will receive the package.

Stopping an advertisement in an emergency

If you receive reports that an advertisement is causing problems on user computers, the most effective way to stop the advertisement is to use the techniques discussed in the "Disabling or Rerunning Advertisements" section earlier in this chapter.

If the advertisement is not an assigned advertisement, and must be initiated by the users, you can also send e-mail or similar broadcasts to the users to advise them to not run the advertised program.

Rerunning an advertisement

If you make changes to a package or program after its advertisements have been run on some clients, you can send an e-mail message to the relevant users to rerun the program. If the advertisement was an assigned advertisement without the option for the users to run the advertisement, you can add a new assignment to the advertisement. The option to rerun an advertisement applies if the advertisement was assigned to run at a scheduled time, instead of on an event (such as logoff). The new assignment will force the advertisement to run again on all the clients in the advertisement's collection.

If you must rerun an advertised program on clients where it failed, you can create a new advertisement to target the same clients or users again. The advertised program will not run again on those clients that successfully ran the program using the first advertisement.

Note

If you delete an advertisement for a package and program, or allow it to expire, and then create a new advertisement for the same package and program, the new advertisement will not run on clients that ran the previous advertisement.

Advertisements with assignments other than **As soon as possible**, **Logon**, or **Logoff** can be rerun on all clients by right-clicking the advertisement, selecting **All Tasks**, and then clicking **Rerun Advertisement**. This creates a new assignment with the current time for the advertisement.

Running an advertised program on a regular basis on clients

To run an advertised program on a regular basis on clients, create an assignment for the advertisement with a recurrence pattern as the schedule.

You might also want to update the distribution points on a regular basis with updated source files. You can do this from the **Data Source** tab of the **Package Properties** dialog box.

Using Windows Installer-based applications with SMS

Windows Installer-based applications maintain a list of sources for the package. If the applications require additional components or replacement copies of files, they can automatically find the original source of the package. The source list includes the location that the application was installed from, which for SMS will be the distribution point. If you remove a distribution point or provide additional distribution points, you might want to add distribution points to the list of sources for the applications.

You can use the following options to add additional resources to the source list:

- Source list entries can be written directly into the Windows Installer package when the package is created.
- Source list entries can be specified on the command line by using the SOURCELIST property. This list is appended to the end of each user's existing source list for the application.
- SMS has a character limit of 255 characters for the command line. If the command line with the source list exceeds this value, use a transform to specify the SOURCELIST property.
- Source list entries can be added at installation time by applying a Windows Installer transform. The transform includes the SOURCELIST property value set to the list of source paths.

You can modify source lists after the application is installed by applying a transform. You cannot modify the source list values after installation if the client is using Windows Installer 1.0.

Advanced Clients verify that .msi packages are Windows Installer packages before attempting to run them. If not, a message is displayed on the client indicating that the file is not a valid Windows Installer package.

Windows Installer packages can have .exe extensions. However, you must use the .msi version of such Windows Installer packages if you want to take advantage of the Windows Installer elevated rights.

For more information about using Windows Installer packages, see the Windows Installer documentation.

Running an advertised program in the user context but with administrator rights

In some cases, you might run an advertised program with administrative credentials but in the user's context, even if the user does not have administrative credentials. This is the case if the setup must perform tasks that require administrative credentials, but it must also perform tasks that can be done in the user's context, such as adding icons to the user's desktop.

Running advertised programs with administrative credentials but in the user's context can be done automatically if the advertised program is a Windows Installer script (.msi file). In addition, the advertised program must be set as requiring administrative credentials and to require user input.

If the advertised program is not a Windows Installer program, installation can be split into two phases that can then be coordinated by using the dependent program feature. The first phase installation program would run under the SMS administrative. The second phase installation program would run under the logged-on user security context to update shortcuts for the logged-on user profile and user-specific registry settings.

Running an advertised program without the users being notified

To run an advertised program without any user intervention, the following criteria must be met:

- The program must be set to run hidden.
- The program must be set to not require any user interaction.
- The program must be set to suppress program notifications.

For more information, see the "Create a New Program" section earlier in this chapter. In addition, the program can be designed to not require any user input.

Estimating how long a package transfer will take

Transferring large packages from site to site, from the site server to a distribution point, or from a distribution point to client, can take a lot of time. This is especially true if the network link is slow, or if the link is already very busy, so that the effective available bandwidth is small. In such cases, it is important for you to estimate how long the package transfer will take. Such estimates will allow you to address two issues:

- You can decide when to start troubleshooting transfers that have not completed.
- You can determine whether a transfer can be accomplished overnight or requires a weekend.

Table 5.7 Approximate Bandwidth for Typical Slow Network Links

Available bandwidth	128 Kbps	28.8 Kbps	9.6 Kbps
Bits/Sec	131,072	29,941	9,830
Bytes/Sec	16,384	3,686	1,229
Bytes/Hour	58,982,400	13,271,040	4,423,680

Using the previous estimates, the following distribution latencies apply.

Table 5.8 Estimated Time to Transfer Packages Over Slow Network Links

Package size	128 Kbps	28.8 Kbps	9.6 Kbps
1 MB	0 D 0:01.04	0 D 0:04.44	0 D 0:14.13
5 MB	0 D 0:05.20	0 D 0:23.42	0 D 1:11.07
10 MB	0 D 0:10.40	0 D 0:47.24	0 D 2:22.13
20 MB	0 D 0:21.20	0 D 1:34.49	0 D 4:44.27
100 MB	0 D 1:46.40	0 D 7:54.04	0 D 23:42.13
400 MB	0 D 7:06.40	1 D 7:36.18	1 D 22:48.53

Using software distribution on computers with terminal services

For clients with Windows Terminal Services (Remote Administration mode or Application Server mode) enabled, software distribution icons and messages are limited to the console session. On clients that are remote controlled using Remote Assistance, Remote Desktop or SMS Remote Control, software distribution icons function regularly. Software Distribution functionality to site systems that have Windows Terminal Services enabled is limited.

On an SMS client, if a package requests a restart, the SMS Advertised Programs Client Agent sends a warning message to users logged on to the system, even if the package was run as a background process. This warning message is not displayed on an SMS client running on Windows 2000 Terminal Services. You should include the Windows 2000 Terminal Services MSG command in any package that reboots clients and is sent to a client running Terminal Services. A package will reboot the system if you have configured the package's program **Properties** dialog box to set **After Running** to either **SMS Restarts System** or **Program Restarts System**.

Expanding the target of advertisements

Advertisements target computers using collections. All the resources within the collection receive the advertisement. If you want more resources to be targeted by the advertisement, you can adjust the collection. You can add additional queries to the collection or additional individual resources.

Software Distribution Best Practices

Applying some best practices to your software distribution procedures will help to ensure success and efficiency. Consider consistently using the following practices:

- Thoroughly test software distributions.
- Distribute software in phases.
- Decrease collection evaluation frequency.
- Distinguish between package distribution and advertisement distribution.
- Make advertisements user-initiated before they are assigned.
- Make advertised programs not require input from users.

Test software distributions

Installing software causes a large number of changes on a computer. Testing packages that you are about to distribute will minimize the risk of problems.

Test your packages on computers that are representative of the computers that will be targeted by your software distributions. In most organizations, computers will vary by computer model, operating system, installed applications, and configuration. Where possible, your tests should include at least one computer that has each combination that will be found on computers targeted by your software distribution.

Ensure that your tests simulate the user experience as closely as possible. Use non-privileged accounts if your users do not have privileges.

Problems caused by a software installation might not be immediately apparent. Verify all aspects of the functionality of tested computers, and allow time for problems to be found.

Testing should begin on computers in a test lab, but later testing should include user computers, or clones of user computers, so that the testing is realistic.

Distribute software in phases

After thorough testing in a lab and on some user computers, there can still be a risk that the software being deployed will cause problems on some computers. Deploy the software in phases, with each phase being larger than the previous phase as your confidence in the package increases. For example, you could deploy to 10 computers on the first day, 100 computers on the next day, 1000 computers on the third, 5000 computers on the fourth, and so on. The initial phases should be a good cross-section of typical computers in your organization, but they should also be to sites where technical specialists are available to help if any problems are found with your package.

Decrease collection evaluation frequency

SMS collections are re-evaluated every 24 hours by default. Frequent updates can be useful for software distribution, because newly discovered computers will quickly receive relevant advertisements. However, in large organizations with many computers and collections, frequent collection evaluation can create considerable workload for the SMS servers. To avoid this, consider decreasing the collection evaluation frequency on some collections.

Distinguish between package distribution and advertisement distribution

In small environments, it is easiest to think of SMS software distribution as one complete process. However, for larger environments, and to minimize the potential for problems, it is best to separate SMS software distribution into at least two processes: package distribution and advertisement distribution.

When you create a package, decide which distribution points the package should be available on, and then add those distribution points to the package. Use the **Package Status** node under the **System Status** node in the SMS Administrator console to ensure that the package is successfully distributed to all target distribution points.

After the package is distributed, you can then start the advertisement process, confident that the package will be available wherever it is needed.

Make advertisements user-initiated before they are assigned

Assigned advertisements will be run on all available computers as soon as the assignment becomes due. Advertisements that must be initiated by users (from **Add or Remove Programs** or other client software distribution programs) will be run when the users run them. User-initiated advertisements will have their workload spread over a longer period of time, minimizing the load on the network and servers at any given time. Also, if there is a problem with a package, you can disable the program as soon as the first users report the problem, preventing other users from being affected by the problem.

Create advertised programs that do not require input from users

If your advertised programs require input from your users, there is a risk that the users might enter the input incorrectly. Another issue is if they provide valid input, but they do it in an inconsistent manner, future troubleshooting or advertised programs might be problematic because of the inconsistencies. To avoid this, ensure that your advertised programs do not require input from users. For more information, see the "Create a Setup Script" section earlier in this chapter.

Collection, package, and advertisement naming

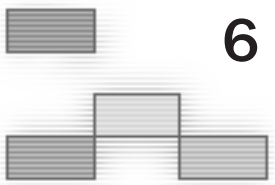
SMS can work properly with collections, packages, or advertisements that have duplicate names. Collections, packages, and advertisements can be created with duplicate names using scripts or tools. When importing collection definitions, the SMS Administrator console does not verify that the collection names are unique. The SMS Administrator console also does not force package and advertisement names to be unique when an SMS administrator creates them. And collections defined at a parent site can have the same name as an already existing collection when they are propagated down to child sites.

However, collections, packages, or advertisements with duplicate names can be confusing to you and other SMS administrators. It can be difficult to find and maintain the correct object, or check its status, if you cannot uniquely identify the object by name. You should ensure that all collections, packages, and advertisements have unique names. If necessary, you could establish a naming convention that includes the site code or creation date to ensure uniqueness.

A naming convention for collections, packages, and advertisements can also make it easier to find the objects if you have many of them. If you have objects that serve similar purposes, you could start their names with a predefined character string that ensures they are listed together when displayed in sorted lists.

CHAPTER 6

Managing Software Updates



Microsoft® Systems Management Server (SMS) 2003 provides a set of tools and procedures that gives system administrators the ability to automate the complex process of managing software updates throughout an enterprise. Chapter 3, "Understanding SMS Features," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* introduces software update management with SMS, including:

- The benefits of using SMS for software update management.
- The major components for managing software updates with SMS.
- The general process of performing software update inventory, distributing software updates, and tracking software update compliance in the enterprise.

This chapter begins with an overview of the software update management process, followed by an overview of each of the software update management components. The chapter then describes the tasks associated with performing a software update inventory, authorizing and distributing software updates to clients, and tracking and maintaining the software update management system.

In This Chapter

- Software Update Management Overview
- Software Update Management Tasks
- Software Update Management Best Practices
- Performance Considerations

Software Update Management Overview

Because software updates are becoming more frequent and important, the task of managing them is critical to the security and the operational health of your enterprise. Using effective software update management techniques has become essential as technology evolves and attackers develop new methods to exploit security vulnerabilities and negatively affect business operations.

Software update management with SMS 2003 is a collection of tools and processes for keeping your SMS client computers current with new software updates that are developed after a software product is released.

About Software Updates

A software update, often referred to as a patch, is a publicly released update to a software product that typically occurs between service packs. Typically, software updates are created and released expeditiously, in reaction to a specific issue. Many, if not most, software updates are released to correct security vulnerabilities. However, software updates also respond to other issues, such as improving performance, extending product functionality, and facilitating product interactions with newly released hardware or software.

Table 6.1 presents the varieties of software updates. In this chapter, the term *software update* is used generically to refer to all of these types of interim product releases.

Table 6.1 Varieties of Software Updates

Term	Definition
Security patch	A publicly released fix that addresses a security issue for a specific product.
Critical update	A publicly released fix that addresses a critical, security related issue for a specific product.
Update	A publicly-released fix that addresses a non-critical, non-security related issue for a specific product; might include new design change requests to add new features or functionality.
Update Rollup	A cumulative set of security patches, critical updates, and updates packaged together for easy deployment. Usually contains all of the software updates for the product since the last service pack or product version release.

About Service Packs

In contrast to a software update, a service pack is an interim product release that is planned and tested over a longer period of time, and consists of a rollup of all software updates (security patches, critical updates, updates, and update rollups or both) that have been released since the last service pack or product release. A service pack can also contain a limited number of customer-requested design changes or features.

Although the SMS 2003 software update management feature does not directly allow you to deploy service packs to your SMS client computers by using the Distribute Software Updates Wizard, you can use SMS software distribution to deploy service packs just as you would deploy any other software. Deploying the latest service pack to SMS client computers is an important part of an effective software update management program, because it:

- Reduces the number of software updates that you must track and manage.
- Reduces the number of updates that your clients must install.
- Reduces the network overhead of the software update management components.
- Decreases the size of software update packages.
- Increases the overall software update compliance in your enterprise.

Service packs are particularly important for software update management because they apply a new baseline for the installed components against which future software updates are applied.

It is imperative that you update the service packs for the systems in your enterprise to defend against any potential security problem. However, in the interim between service packs, the most important thing you can do to maintain a secure system is to make sure that the computers in your enterprise are running the most current security updates.

Challenges in Managing Software Updates

Patching and maintaining managed resources is a reality of networked, distributed computing. An effective software update management process is necessary to maintain operational efficiency, overcome security issues, and maintain the stability of the network infrastructure. However, because of the changing nature of technology and the continual appearance of new security threats, the task of effective software update management can be challenging.

The main challenge in managing security updates is determining which of the many available software updates are appropriate to the requirements and potential security problems of your managed resources and finding the balance that is appropriate for your enterprise.

- Some updates are critical and require immediate action to protect your systems, data, or network infrastructure. For example, the updates that address risks from newly discovered exploitations, viruses, and worms are considered critical updates.
- Some updates can be useful, can increase performance or stability, or can make the end-user experience better, but they might not be considered critical to the safety of your enterprise.

- Some updates might not be necessary to your enterprise and you can ignore them.
- Some updates could create problems (for example, break other line-of-business applications) for your enterprise if you used them.

To keep your enterprise secure, you must establish processes for:

- Receiving information about the latest software updates and vulnerabilities.
- Auditing your enterprise for applicable software updates.
- Assessing and authorizing available software updates.
- Deploying authorized software updates within your enterprise in a timely, accurate, and efficient manner.
- Tracking update deployment across your enterprise.

Software Update Management Guidelines

To learn how to determine which updates are critical, useful, irrelevant, or harmful to your enterprise and to create a software update management process for your enterprise, you can do several things:

- Be familiar with the current state of the resources in your enterprise. This includes knowing:
  - The computers in your enterprise.
  - Operating systems and versions running on each computer.
  - Software updates in use on each computer (service pack versions, software updates, and other modifications).
  - The function each computer performs in your enterprise.
  - The applications and programs running on each computer.
  - Ownership and contact information.
  - The assets present in your environment and their relative value to determine which areas need the most protection.
  - Known security problems and the processes your enterprise has for identifying new security issues or changes in security level.
  - Countermeasures that have been deployed to secure your environment.

  You should update this information regularly, and it should be readily available to those involved in your software update management process.

- Read the white papers listed in Table 6.2 for information and guidelines for establishing a software update management process in your enterprise by using SMS and the Feature Pack tools. These white papers are available at the Microsoft Solutions for Management Web site at http://www.microsoft.com/solutions/msm.

Table 6.2 Software Update Management White Papers

Title	Definition
Patch Management Using SMS/Architecture Guide	Provides architectural guidance for deploying software updates, service packs, and Quick Fix Engineering (QFE) fixes by using SMS and the Feature Pack tools.
Patch Management Using SMS/Deployment Guide	This white paper provides conceptual information, best practices, and detailed procedures that are related to distributing and managing software updates by using SMS, including essential maintenance tasks and team role responsibilities.
Patch Management Using SMS/Operations Guide	This document provides operational guidance for deploying software updates, service packs, and QFE fixes by using SMS. It describes the daily, weekly, monthly, and as-needed tasks that have to be completed to deploy patches into a live production environment.

1. Be informed about the latest security developments and technology. You can be informed by reading, using Web sites, and joining newsgroups to get the latest information.
2. Use the SMS software update management components to streamline and automate some of the functions associated with security update inventory, deployment and management tasks, such as:
   - Conducting an audit of applicable and installed security updates for all the computers in your enterprise.
   - Authorizing and deploying the updates to the appropriate computers.
   - Tracking the inventory and update installation status and progress for all the computers in your enterprise.

How Software Update Management Works

Chapter 3, "Understanding SMS Features," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* provided a general introduction to the software update management process with SMS 2003. The sections that follow provide a more detailed description of the software update management components and their function.

Basic Components Functionality

When the scan component of the software update inventory tools runs on client computers, it creates a new class in the WMI schema for that computer named **Win32_Patchstate**. The scan component examines the registry of the client computer and compares the information contained there to the current catalog of known software updates from Microsoft (Mssecure.xml for security updates and Invcif.exe for Microsoft Office). When the scan component finds an update that is either installed or not yet installed but applicable, it adds an instance to the **Win32_Patchstate** class for that update. This information then propagates up to the SMS site database through the standard SMS hardware inventory process.

Periodically (weekly by default), the synchronization component of the software update inventory tools downloads the latest software update catalog and the latest versions of the scan components from the Microsoft Downloads center and distributes these to SMS distribution points, from which the updated components are distributed to SMS client computers.

When the administrator runs the Distribute Software Updates Wizard from the SMS Administrator console of a site server, several things happen:

- The wizard connects to the SMS site database and obtains the latest version of the software update inventory data contained in the hardware inventory records for the type of software updates currently being managed (for example, Security or Office).
- The wizard displays that list to the administrator, allowing the administrator to select and configure the software updates for the current package.
- If the administrator is creating a new package, the wizard creates a package and program object for the software update type in the specified package source folder.
- When the administrator authorizes software updates, the wizard creates a software updates installation list (PatchAuthorize.xml) and adds the information about the selected software update to this list. This list is also stored in the package source folder.
- The wizard copies the Software Updates Installation Agent (PatchInstall.exe) to the package source folder and creates a program object that contains the configurable settings that the administrator specifies the agent should use when it installs the updates on client computers.
- If directed by the administrator, the wizard also creates an advertisement for distributing the software update package to the specified client collection.

When the advertisement for the software update package runs on SMS client computers, the Software Updates Installation Agent runs with the configuration options that were specified by the administrator in creating the program for the package. When software updates are installed, either automatically or as requested by the user of the computer (depending on program settings), the agent first runs the scan component for the relevant software updates inventory tool to determine which of the software updates to be installed are applicable and missing from the client computer. If the destination computer is running the SMS Advanced Client, the agent can also be configured to run a local notification and scheduling process on the client computer (the persistent notification icon). For more information about this icon, see the “Software Update Management Advanced Features” section later in this chapter.

Each action taken by the Software Updates Installation Agent is logged, and it is also recorded in the form of SMS status messages. These status messages provide a near-real-time record of the compliance level of the computer with respect to the software updates that are contained in the package. In particular, software updates that have been installed, but which are not yet in effect pending a system restart, are recorded as such.

The above description covers the basic operation of the software update management components. However, several new advanced features have been added to the software update inventory tools for SMS 2003 which allow you to perform more complex tasks. These features are described in the following section.

Underlying Technology

The software update inventory tools use the following existing technology to provide you with a better software update management solution:

Security Patch Bulletin Catalog (MSSecure.XML) This is the security updates database that the Microsoft Baseline Security Analyzer (MBSA) and the Security Update Inventory Tool use to determine which security updates are installed on your computers and which are applicable. The Security Update Inventory Tool synchronization component automatically downloads the latest version of this database on a regular basis and distributes it to the computers in your enterprise by using SMS distribution points. For more information about MSSecure.XML, see the Microsoft Web site at http://www.microsoft.com/technet.

Microsoft Baseline Security Analyzer (MBSA) MBSA runs on Microsoft Windows® operating systems and scans for applicable security updates in the operating system, and in other products, such as Microsoft Internet Explorer, Microsoft Windows Media® Player, and Microsoft SQL Server™. The Security Update Inventory Tool includes MBSA technology in its scan component. The Security Update Inventory Tool synchronization component automatically downloads the latest version of this tool on a regular basis and distributes it to the computers in your enterprise by using SMS distribution points. For more information about the MBSA, see the Microsoft Web site at http://www.microsoft.com/technet/security/tools/Tools/mbsahome.asp.

Microsoft Office Update Tool (Invcm.exe) The Microsoft Office Inventory Tool for Updates uses the Microsoft Office Update Tool with the Microsoft Office Update Database (Invcif.exe) to analyze your client computers for applicable updates to Microsoft Office programs. The data gathered by the Microsoft Office Update Tool is then converted into a format that is compatible with the SMS site database. The Microsoft Office Inventory Tool for Updates synchronization component automatically downloads the latest version of the Microsoft Office Update Tool on a regular basis and distributes it to the computers in your enterprise by using SMS distribution points.

Microsoft Office Update Database (Invcif.exe) This is the database of software updates that the Microsoft Office Update Tool and the Microsoft Office Inventory Tool for Updates use to determine which office updates are installed on your computers and which are applicable. The Microsoft Office Inventory Tool for Updates synchronization component automatically downloads the latest version of this database on a regular basis and distributes it to the computers in your enterprise by using SMS distribution points. For more information about the Microsoft Office Update Tool, see http://support.microsoft.com?kbid=312982.

Software Update Management Advanced Features

The following advanced features are included with the software update management feature in SMS 2003.

Persistent Notification

The persistent notification icon is a feature that allows a user on a computer that is running the SMS Advanced Client to receive notifications and schedule software update installations independent of the software update advertisement. This allows for better compliance by allowing users to install updates at their convenience, and it reduces system load because the advertisement does not have to be scheduled as often.

If this feature is enabled by the SMS administrator for a software updates program or package, an icon appears in the notification area (formerly called the system tray) whenever a user is logged on and there are pending, uninstalled software updates. When the computer is in compliance, the notification area icon does not appear.

Users can use the notification area icon to:

- Check for upcoming installations.
- Schedule installations and restarts to occur at convenient times of the day.
- Install software updates immediately.

If the computer is running the Legacy Client, the persistent notification settings are ignored. You can enable this feature for a package or program on the third **Configure Installation Agent Settings** page of the Distribute Software Updates Wizard.

Unattended Software Update Installation

Unattended software update installations are installations that occur without notification or user interaction. No notification icon appears in the system tray, and users with insufficient credentials cannot terminate the process in Task Manager. This feature is useful for pushing critical software updates quickly through the enterprise and can be effective in locked-down installations or situations where enterprise policy dictates strict compliance rules.

You can enable unattended software update installations for a package or program through settings on the **Configure Installation Agent Settings** pages of the Distribute Software Updates Wizard. For more information, see the "Configure Software Updates Installation Agent Settings" section later in this chapter.

Firewall Authentication Support

Because the synchronization component of the software update inventory tools requires access through the firewall to the Internet, this can create problems in enterprises with stringent firewall policies.

You can now run the synchronization component to obtain catalogs of software updates in an automated, unattended way, even through a firewall that requires authentication of a domain user account. You can also optionally specify a user name and password of an account that is authenticated through the firewall, in addition to the IP address of a specific proxy server. For more information, see the "Configure the Synchronization Host" section later in this chapter.

Scheduled Installations

To accommodate the special requirements of servers, which often can be maintained only at certain hours on certain days, you can now configure the Distribute Software Updates Wizard and the Software Updates Installation Agent to limit the time that a software update is installed to a specific time period. Outside of this time period, no installation is performed. If the SMS client is offline during the time period when the advertisement is scheduled, the restricted time period prevents the SMS client from attempting to catch up and apply the software updates at the wrong time.

Dynamic Package Configuration

You can use dynamic package configuration to create multiple program objects for the same package. This allows you to distribute one package with multiple installation parameters, so that you can conditionally install the package to different collections according to criteria you define. For example, you can create one program for workstations that are running the Legacy Client, another for mobile users that are running the Advanced Client (with, for example, a less frequent advertisement schedule) and a third program for servers on which system restarts are automatically suppressed and a scheduled installation is specified.

You can also attach a different software updates authorization list to each program in the package, so you can, for example, add a newly released software update to your production package and distribute it only to your test collection.

Reference Computer Inventory Template

Because the Distribute Software Updates Wizard does not list a software update for approval until the update has been requested by at least one client computer, there might be some delay between the time a software update becomes available and the time it is approved for distribution. You can use this feature to specify a reference computer to generate baseline software update templates, which speeds authorization, package administration, and package deployment.

Software Update Management Tasks

There are three main tasks you perform in managing software updates Each task is divided into several subtasks:

- **Preparing for software update management**

  This is a one-time step that involves downloading and running the installer program for the software update inventory tool on the site server and then distributing the tool components to the destination client computers.

- **Authorizing and distributing software updates**

  This is a recurring task that you perform as often as is required by the size and rate of change of the sites you are administering.

- **Tracking software update compliance**

  In this task you monitor the software update installation process, check compliance levels for critical updates and troubleshoot software update installation problems.

These tasks are described in detail in the following sections.

Preparing for Software Update Management Tasks

Preparing a site for software update management is a separate process that you can perform after you deploy SMS 2003 in your enterprise. For best results, and to help protect your network against security vulnerabilities, it is recommended that you deploy the software update management feature soon after your SMS hierarchy is set up and configured.

Preparing for software update management involves the following tasks:

- Review the system requirements for the software update management components.
- Prepare the test environment.
- Prepare the production environment.
- Deploy the software update inventory tools by:
  1. Planning the deployment.
  2. Downloading and running the installer on the site server.
  3. Creating the necessary collections, programs, and advertisements.
  4. Configuring the synchronization host.
  5. Verifying the installation.
  6. Performing a test inventory.
  7. Distributing the tools to client computers.

These preparatory tasks are described in the following sections.

Task 1: Review the System Requirements for the Software Update Management Components

The software update management feature of SMS 2003 consists of a series of interacting components, some of which are installed by default when you install the SMS Administrator console on the site server. Other components require a separate download and installation. Table 6.3 lists the software update management components and their installation details.

Table 6.3 Installation Details for the Software Update Management Components

Component	Installation
Distribute Software Updates Wizard	Installed by default with SMS Administrator console.
Software Updates Installation Agent	Installed by default with SMS Administrator console.
Software updates reports	Installed by default with SMS Administrator console.
Security Update Inventory Tool	Available by download from Microsoft Downloads Center. Separate installation on site server.
Microsoft Office Inventory Tool for Updates	Available by download from Microsoft Downloads Center. Separate installation on site server.

The "Getting Started" chapter of *the Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* outlines the system requirements for site servers and other site systems that are running SMS 2003. These system requirements are the same for all of the software update management components that are installed by default when you install SMS 2003.

The following sections outline the system requirements for the software update inventory tools (Security Update Inventory Tool and the Microsoft Office Inventory Tool for Updates.)

System Requirements for the Software Update Inventory Tools

Each of the software update inventory tools is delivered in an installer package (for example, the Security Update Inventory Tool Installer or the Microsoft Office Inventory Tool for Updates Installer). When you run this installer package on the SMS site server, it automatically builds the packages, collections, and advertisements that are needed to deploy the other tool components within your site.

Each installer package contains two components:

- **Scan component** (S_scan.exe or O_scan.exe)

  This component runs on the SMS client computers in your enterprise and carries out automated, ongoing scans for installed or applicable (not yet installed) updates. It then converts the gathered data into SMS inventory data.

- **Synchronization component** (Syncxml.exe for both the Security Update Inventory Tool and the Microsoft Office Inventory Tool for Updates)

  This component runs on a single computer that has an Internet connection. It periodically checks the Microsoft Downloads Center Web site and downloads the latest security update bulletin catalog. It then uses SMS distribution points in your site to send the latest version of the catalog to SMS client computers.

Note

The Security Update Inventory Tool and the Microsoft Office Inventory Tool for Updates are separate tools; each tool can be installed and deployed without the other.

System requirements for the Security Update Inventory Tool

The Security Update Inventory Tool is packaged in an installation program named SecurityPatch_*xxx*.exe, where *xxx* is the locale extension for the package. Run this installation program on the site server that is at a level in the SMS hierarchy that contains all of your destination clients for security update scans.

Table 6.4 shows the installation requirements for the installation program and the two client components.

Table 6.4 Installation Requirements for the Security Update Inventory Tool

Component	File name	Runs on	Internet Explorer version	Platform	Other dependency
Installer	SecurityPatch_*xxx*.exe	Site server	Not applicable	Microsoft Windows NT® 4.0 SP6a or later	MSXML 3.0, SP4[1]
Scan	S_scan.exe	SMS client	5.0 or later	Windows NT 4.0 SP4 or later	MSXML 3.0, SP4[1]
Synchronization	Syncxml.exe	SMS client[2]	Not applicable	Windows NT 4.0 SP6a or later	MSXML 3.0, SP4[1]

[1] See the "About the Microsoft XML dependency for the software update inventory tools" section later in this chapter.

[2] See the "Preinstallation requirements for the synchronization component" section later in this chapter for the special requirements for this SMS client computer.

System requirements for the Microsoft Office Inventory Tool for Updates

The Microsoft Office Inventory Tool for Updates is packaged in an installation program named OfficePatch_*xxx*.exe, where *xxx* is the locale extension for the package. Run this installation program on the site server that is at a level in the SMS hierarchy that contains all of your destination clients for Office software update scans.

Table 6.5 shows the installation requirements for the installation program and the two client components. Note that the minimum supported client operating system requirement is different from that of the Security Update Inventory Tool.

Table 6.5 Installation Requirements for the Microsoft Office Inventory Tool for Updates

Component	File name	Runs on	Internet Explorer version	Platform	Other dependency
Installer	OfficePatch_*xxx*.exe	Site server	Not applicable	Windows NT 4.0 SP6a or later	MSXML 3.0 SP4[1]
Scan	O_scan.exe	SMS client	5.0 or later	Windows NT 4.0 SP5 or later	MSXML 3.0 SP4

(continued)

Table 6.5 Installation Requirements for the Microsoft Office Inventory Tool for Updates *(continued)*

Component	File name	Runs on	Internet Explorer version	Platform	Other dependency
Synchronization	Syncxml.exe	SMS client[2]	Not applicable	Windows NT 4.0 SP6a or later	MSXML 3.0 SP4

[1] See the "About the Microsoft XML dependency for the software update inventory tools" section later in this chapter. Also, see the System Requirements section of the product release notes for the most current information about the Microsoft XML version.

[2] See the "Preinstallation requirements for the synchronization component" section later in this chapter for the special requirements for this SMS client computer.

About the Microsoft XML dependency for the software update inventory tools

The software update inventory tool scan components (Security Update Inventory Scan Tool and Microsoft Office Inventory Scan Tool for Updates) both require MSXML, version 3.0 SP2 to run on SMS client computers. If this application is not found, the scan components install it. The tools detect older versions by looking for Msxml3.dll having a version earlier than 8.40.9419.0 in the %*Windir*%\system32 folder of the SMS client computer.

If you have applications that are not compatible with this version of MSXML and want to bypass this upgrade, you can preinstall the Msxml3.dll and Msxml3r.dll files on client computers before you deploy the inventory scan programs, or you can change the scan tool program command-line by using the following procedure. This prevents the automated upgrade to MSXML 3.0 SP4 if it is not required in your environment.

Important

Versions of MSXML that are earlier than version 3.0 SP2 have not been extensively tested for use by the scan component and are not recommended.

To suppress the MSXML upgrade on the client computer

1. In the SMS Administrator console on the site server where you ran the software update inventory tool installer, navigate to the scan tool package.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Packages
   └ package
```

2. In the results pane, right-click the program you want to modify, and then click **Properties**.

3. Change the command-line to:

```
s_scan.exe /s /cache /noxml
```

– Or –

```
O_scan.exe /s /cache /noxml
```

Preinstallation requirements for the synchronization component

The synchronization component of the software update inventory tools (Syncxml.exe) is installed on an SMS client computer with access to the Internet (the synchronization host). You specify this computer when you run the installer program for a software update inventory tool.

The synchronization component performs the following tasks:

- Connects to the Microsoft Downloads Web site through the firewall.
- Attempts to download the latest software update catalog into the package source folder of the SMS software update inventory tool package.
- Optionally performs a dynamic update of the distribution points after the download is complete.

To perform these tasks, the synchronization component requires:

- Internet access with the HTTP 1.1 protocol enabled through the firewall.
- Read/write access to the package source folder.
- Access to the package object (if the synchronization component is configured to dynamically update the distribution points).

For more information about configuring the synchronization component, see the "Configure the Synchronization Host" section later in this chapter.

Avoiding problems caused by FAT formatted systems

You should be aware, when preparing your client computers for running the software update inventory tools, that the FAT (file allocation table) file system is inherently not secure. Software update solutions that involve FAT file systems cannot and do not match the level of security that is available from an NTFS file system format. For example, clients that are running NTFS can safely run the software update inventory scan from a secure local cache (controlled by the scan component **/cache** parameter).

If an SMS client is running on a computer that has a FAT file system on a system partition, the SMS 2003 software update inventory tools still use a local cache to run the software update inventory scan (under the **/cache** parameter), in the same way that an NTFS system would, for performance reasons. However, that cache is inherently not secure under a FAT system and does not become secure until the system partition has been converted to NTFS, after which it is automatically accessible only by system administrators.

It is recommended that you convert clients running FAT file systems to NTFS file systems as soon as possible if the computer can support it. Common reasons for having a FAT system include dual-booting to Microsoft Windows 98, or to another operating system that requires a FAT formatted system.

To learn how to convert a file system from FAT to NTFS, refer to the help available by typing **convert /?** at the command prompt.

Task 2: Prepare the Test Environment

This section describes the operating systems and settings that are necessary to create a minimum configuration of an SMS site to use while you are testing or evaluating the software update management components.

For example, if your enterprise uses Microsoft Windows 2000, Microsoft Windows XP, and Microsoft Windows NT 4.0, you will need a minimum of one computer for each configuration.

In addition, you need computers that have other crucial line of business applications running on them (for example, accounting or sales tracking software).

When configuring a test collection, you should also account for variation in hardware within your enterprise (desktop versus laptop computers) and hardware configurations (low memory versus multiprocessor servers).

Client Requirement

One client is sufficient for minimum test purposes. However, if you want to have a representative sample of how the tools will work with all of the systems used in your enterprise, it is recommended that you have at least one Advanced Client and one Legacy Client for each representative configuration in your environment.

For example, if you have computers that are running Windows 2000 SP3 and Windows NT 4.0 SP6a, you should have a client computer for each of these operating systems in your test configuration.

If you do not currently use a certain operating system (for example, Windows XP) in your enterprise, but you plan to use it in the future, it is recommended that you add a computer that is running that system to your test configuration. This allows you to become familiar with how the software update management components and software updates work with the operating system before you deploy it in your enterprise.

Setting up this type of extended client test configuration allows you to become familiar with software update management in many different ways. By using more than one operating system, you can:

- Review the specific software updates that Microsoft has published for those operating systems.
- Start to get familiar with update management practices for each system.
- Learn how the updates work with different operating systems, in a controlled environment.
- Learn how to find information about specific updates for specific operating systems when you need it.

For more information about configuring SMS client computers, see Chapter 4, “Understanding SMS Clients,” in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Hardware Inventory Settings

The software update inventory tools use hardware inventory to create an inventory of installed and applicable software updates on your client computers.

By default, the hardware inventory function is disabled on the SMS primary site to reduce system overhead. To set up your test system, you must enable the hardware inventory function and configure the inventory frequency.

The default frequency for SMS hardware inventory is an interval of seven days. However, for test purposes, to speed the process of becoming familiar with the software update inventory tools, you can increase the frequency of the inventory, perhaps running it daily, or even every few hours.

Note

The above hardware inventory setting suggestions are for test purposes only. The actual frequency with which you run the hardware inventory in a full-scale deployment of the tools depends on the needs of your enterprise and performance considerations associated with the generation of additional hardware inventory data.

For more information about configuring the Hardware Inventory settings, see Chapter 2, "Collecting Hardware and Software Inventory."

For more information about specific performance issues associated with these tools, see the "Performance Considerations" section later in this chapter.

Software Distribution Settings

Some of the software distribution settings for SMS might conflict with those of the software update management components and could cause confusion. To prevent this possibility, configure the following settings on the SMS primary site:

- Turn off the site-wide countdown for assigned programs.

  Both SMS software distribution and the software update inventory tools have countdown features for assigned programs. To prevent duplicate countdowns, disable this feature on the SMS primary site; the countdown features provided by the software update management components can be changed or eliminated as needed.

- Turn off the notification for software distribution activity.

  Both SMS software distribution and the software update inventory tools contain a notification feature that tells you when software distribution activity is occurring. To prevent confusion caused by duplicate notifications, you can choose to disable this feature on the SMS primary site.

- Modify the Advertised Program Client Agent polling interval.

  By default, the software distribution system on a client computer checks for software distribution activity every hour. For test purposes, to avoid unnecessary delays, you can increase the polling frequency to an interval of five or ten minutes.

> **Note**
> In a test environment, a short polling interval causes few system resource usage problems. However, when deploying the tools to a larger system, the polling interval should be increased, for example, to a four-hour interval to prevent performance problems.

For more information about configuring the SMS software distribution settings, see Chapter 5, "Distributing Software."

For more information about specific performance issues associated with these components, see the "Performance Considerations" section later in this chapter.

Task 3: Prepare the Production Environment

The settings and configurations that are suggested in the "Prepare the Test Environment" section earlier in this chapter help you become familiar with the software update management components and how they work with your SMS system in a small-scale test environment.

However, when you deploy these components on a larger scale, you should be aware that these settings and configurations must change, or performance issues could result. The reason for this is that as the scale of software update management component deployment increases, so do the demands on your system.

Hardware inventory size, network usage, CPU usage, and disk capacity requirements all increase proportionately to the size of your deployment. Also, the settings you configure for SMS and the software update management components influence the impact of the processes on your system. For example, if you were to increase the advertisement schedule for the software update inventory tool scan process from a weekly to a daily interval, the system overhead caused by that activity would increase from approximately 5 percent to 15 percent overall.

For larger scale deployment, the following SMS settings are suggested for use with the software update management components:

- Configure the SMS Hardware Inventory cycle to occur weekly.
- Configure SMS software distribution settings as follows:
  - Turn off the site-wide countdown for assigned programs.
  - Turn off the notification for software distribution activity.

As mentioned in the "Software Distribution Settings" section earlier in this chapter, both SMS software distribution and the software update management components have countdown and notification features for assigned programs. To prevent duplicate countdowns and notifications, disable these features for software distribution on the SMS primary site. The countdown and notification features that are provided by the software update management components can be changed or eliminated as needed.

Note

There might be other software distribution practices occurring in your enterprise that use the SMS countdown and notification features. You should review these before you make the recommended changes; however, that review should also take into account the countdown and notification features that are provided by the software update management components.

Task 4: Deploy the Software Update Inventory Tools

The following is a summary of the steps that are required to deploy the software update inventory tools (Security Update Inventory Tool and Microsoft Office Inventory Tool for Updates). Each step is fully discussed in the subsequent sections. For more information and the most current information about installing and using the software update inventory tools, see the Help file that is installed with each tool.

1. Plan the deployment.
2. Download and run the installer on the site server.
3. Create the necessary collections, programs, and advertisements.
4. Configure the synchronization host.
5. Verify the installation.
6. Perform a test inventory.
7. Distribute the tools to client computers.

Plan the Deployment

Before deploying the software update inventory tools in a production environment, you should:

- Determine the types of software updates to be managed.
- Plan the strategy for collections and program advertisements.
- Plan the synchronization task and schedule.
- Perform a test deployment.

Determine the types of software updates to be managed

There are two software update types that you can manage with the SMS 2003 software update inventory tools:

- Security
- Office

Security updates are updates to Microsoft operating systems and other systems software. If you want to manage security updates, begin by deploying the Security Update Inventory Tool.

Office updates are software updates to Microsoft Office software. If you want to manage Office updates, begin by deploying the Microsoft Office Inventory Tool for Updates.

Note that you can install either tool independent of the other.

Plan the strategy for collections and program advertisements

When you initially install the Security Update Inventory Tool or the Microsoft Office Inventory Tool for Updates on the site server, the installer program can automatically create the necessary collections, packages, programs, and advertisements you must have to deploy the tool component to SMS client computers in your enterprise. These default objects are designed to assist you in deploying the software update inventory tools in your enterprise and to work together with the other software update management components, such as the Distribute Software Updates Wizard. However, in some cases these default objects are not sufficient to meet the needs of you enterprise. In this case, it is recommended that you allow the installer program to create the default objects for you automatically, and then create your own collections and create or modify the other objects you must have when you finish testing the tools. For a list of the considerations you should take into account when creating or modifying these objects, see the "Software Update Management Best Practices" section later in this chapter.

The default objects that are created for the software update inventory tools are listed in Table 6.6. You supply the root *toolname* when you run the installer program for the tool on the site server. It is important to select a *toolname* that easily identifies the tool you are installing and distinguishes it from other instances of the tool that might be running in other areas of the site.

Table 6.6 Software Update Inventory Tool Default Objects

Object	Purpose
Collections	
Scan tool collection *toolname* (*sitecode*)	The main collection for distributing the scan component to SMS client computers. After installation is completed, the Security Update Inventory Tool package is advertised to this collection. Initially after installation, this collection is restricted by a query limitation to contain the computers that are in the pre-production collection described below.

(continued)

Table 6.6 Software Update Inventory Tool Default Objects *(continued)*

Object	Purpose
Scan tool (pre-production) collection *toolname* (*sitecode*) pre-production	You can use this collection to test the software update packages that you create with the Distribute Software Updates Wizard. The collection is defined with a direct membership rule that contains the computer you specified as the test computer when you ran the Security Update Inventory Tool Installer.
Synchronization component collection *toolname* (*sitecode*) Sync host	If you specified a computer to run the synchronization component when you ran the installer for the Security Update Inventory Tool or the Microsoft Office Inventory Tool for Updates, this collection is created. It is defined by a direct membership rule that contains only the computer you specified, and it receives advertisements from the synchronization program of the scan component package.
Package	
Software update inventory tool package *toolname* (*sitecode*)	The main package for distributing Security Update Inventory Tool client components to SMS client computers. The package node contains subnodes for access accounts, distribution points, and programs. Under the **Programs** subnode, the distribution package contains the three programs described below by default:
Programs	

Scan component program
toolname (*sitecode*)
	The generic program for running the scan component on SMS client computers in a production environment. By default, this program runs the scan component with the following command line for the Security Update Inventory Tool:
	`s_scan.exe /s /cache`
	Or, for the Microsoft Office Inventory Tool for Updates:
	`o_scan.exe /s /cache`

Scan component expedited program
toolname (*sitecode*) expedited
	A special program for running the scan component on SMS client computers in an expedited manner in a test environment. For performance reasons, you should not use the program in a production environment.
	By default, this program runs the scan component with the following command line for the Security Update Inventory Tool:
	`s_scan.exe /s /cache /kick`
	Or, for the Microsoft Office Inventory Tool for Updates:
	`o_scan.exe /s /cache /kick`

(continued)

Table 6.6 Software Update Inventory Tool Default Objects *(continued)*

Object	Purpose
Synchronization component program *toolname* (*sitecode*) Sync	This program runs the synchronization component on the synchronization host. By default, this program runs the synchronization component (Syncxml.exe) with the following command line for both the Security Update Inventory Tool and the Microsoft Office Inventory Tool for Updates: `syncxml.exe /s /site sitename /code sitecode /target packagelocation /package packagename`
Advertisements	
Scan component advertisement *toolname* (*sitecode*)	Advertisement for distributing the scan component to client computers. Scheduled to run every seven days by default. By default, this advertisement runs the standard (not expedited) scan component program.
Synchronization component advertisement *toolname* (*sitecode*) Sync	Advertisement for the synchronization component. Scheduled to run every seven days by default.

Plan the synchronization task and schedule

Each of the software update inventory tools contains a synchronization component. This component runs on a designated SMS client computer that has access to the Internet and is configured by an advertisement to run the synchronization task at a regular interval. The purpose of the synchronization task is to keep the scan components current with the latest software update catalogs from Microsoft.

Because the synchronization task requires authenticated access through the firewall to the Internet and also requires access to the package source folder, there are several important points to take into account when you are planning for this component, such as:.

- Whether to run the synchronization component in attended mode or unattended mode.
- How frequently and when to schedule the synchronization task. For more information, see the “Scheduling: Best Practices” section later in this chapter.
- How to enable access to the package source folder. The easiest way is to install the synchronization component and the package source folder on the same computer.

If you plan to run the synchronization host in unattended mode, you must do the following:

- Ensure that the source directory for the scan component package is located on the synchronization host. This is because the SMSCliToknLocalAcct& account does not have permissions to update this directory over the network.
- The firewall for the synchronization host must allow anonymous access, or you must provide the user name and password of an authenticated user for the synchronization task to use.

For more information about configuring the synchronization component, see the "Configure the Synchronization Host" section later in this chapter.

Download and Run the Installer on the Site Server

The following sections give you general instructions and notes for running the installer program for each of the software update inventory tools.

Installing the Security Update Inventory Tool

The Security Update Inventory Tool is packaged in an installation program named SecurityPatch_*xxx*.exe, where *xxx* is the locale extension for the package. Run this installation program on the site server that is at a level in the SMS hierarchy that contains all of your clients that are targeted for security update scans.

Before you run the Security Update Inventory Tool Installer you must:

- Know the SMS site server computer name and site code.
- Have package creation credentials.
- Have collection and advertisement creation credentials, if you choose to allow the installer program to create these objects (recommended).
- Be ready to provide the NetBIOS name of an existing SMS client computer with Internet access, if you choose to deploy the synchronization component by using the installer program.

In addition, you should review the preinstallation requirements for the Security Update Inventory Tool.

The following sections provide general information about the options available on some of the pages of the Security Update Inventory Tool Installer. For more detailed steps, see the documentation for the tool available at the Microsoft Downloads Web site at http://www.microsoft.com/smserver/downloads.

► To run the Security Update Inventory Tool Installer

1. Download the Security Update Inventory Tool Installer for SMS 2003 from http://www.microsoft.com/smserver/downloads.
2. Run the Security Update Inventory Tool Installer on the site server.

3. Step through the installation wizard to install the tool components, noting the following:

   - The **Scan Tool Download** page of the wizard prompts you to download the security bulletin file (Mssecure.cab), which is a required dependency of the Security Update Inventory Tool.

     Note

     If you are installing the Security Update Inventory Tool on a computer that does not have Internet access, you can download the file manually from http://www.microsoft.com/smserver/downloads and then copy it to the installation folder of the Security Update Inventory Tool (the default folder is C:\Program Files\Security Update\1033. You might be required to create this folder).

   - The **Distribution Settings** page of the installation wizard allows you to configure the default objects that are created by the installation wizard. These objects include packages, programs, collections, and advertisements that you must have to deploy the Security Update Inventory Tool to your SMS client computers. For more information about these default objects, see Table 6.6.

     On this page you can also specify whether or not you want setup to assign the distribution package to all of the distribution points in your site. If you choose not to have this done, the package is not assigned to any distribution points, and you can use the standard package management features of the SMS Administrator console to assign the package to the distribution points of your choice.

     The last part of this page prompts you to assign a name to these objects. You should choose a name that allows you to clearly identify the tool and software update type you are installing, and that allows you to distinguish this instance of the tool from instances that are installed on other sites in the hierarchy.

     Caution

     Renaming these objects after they are created might cause some parts of the software update inventory process to fail.

   - On the **Database Updates** page of the installation wizard, specify the name of an Internet-connected SMS client computer to run the Security Updates Sync Tool task. The computer that you specify here is the synchronization host, and it requires authenticated Internet access through the firewall. For more information about configuring synchronization component access through the firewall, or for installation on sites without Internet access, see the "Configure the Synchronization Host" section later in this chapter.

     Setup places the specified computer into a collection and creates a weekly advertisement to download, install, and distribute updated versions of the synchronization component and database. By default, the advertisement is assigned on a weekly basis within the security context of the user who is currently logged on and running the Installer.

If you do not supply a computer name and leave the text field blank, setup creates only the synchronization component program, but not the collection or advertisement.

- On the **Test Computer** page of the installation wizard, specify a test computer to be added to the test collection that setup creates (the pre-production collection). By default, the test collection is specified as the value of the **Limit to collection** property of the main collection. In most cases you will want to add more computers to this test collection after you complete the installation process. For more information, see the "Task 2: Prepare the Test Environment" section earlier in this chapter.

Installing the Microsoft Office Inventory Tool for Updates

The Microsoft Office Inventory Tool for Updates is packaged in an installation program named OfficePatch_*xxx*.exe, where *xxx* is the locale extension for the package. Run this installation program on the site server that is at a level in the SMS hierarchy that contains all of your targeted clients for Office update scans.

Before you run the Microsoft Office Inventory Tool for Updates Installer you must:

- Know the SMS site server computer name and site code.
- Have package creation credentials.
- Have collection and advertisement creation credentials, if you choose to allow the installer program to create these objects (recommended).
- Be ready to provide the NetBIOS name of an existing SMS client computer with Internet access, if you choose to deploy the synchronization component using the installer program.

In addition, you should review the preinstallation requirements for the Microsoft Office Inventory Tool for Updates.

The following notes provide general information about the options that are available on some of the pages of the Security Update Inventory Tool Installer. For more detailed steps, see the documentation for the tool available at the Microsoft Downloads Web site at http://www.microsoft.com/smserver/downloads.

► To run the Microsoft Office Inventory Tool for Updates Installer

1. Download the Microsoft Office Inventory Tool for Updates Installer for SMS 2003 from http://www.microsoft.com/smserver/downloads.
2. Run the Microsoft Office Inventory Tool for Updates Installer on the site server.

3. Step through the installation wizard to install the tool components, noting the following:
   - The **Office Update Inventory Tool** page prompts you to download the Office Update Inventory files (Invcif.exe and Invcm.exe), which contain the latest tool and catalog for scanning Microsoft Office.

     Note

     If you are installing the Microsoft Office Inventory Tool for Updates on a computer that does not have Internet access, you can download the file manually at http://www.microsoft.com/smserver/downloads and then copy it to the installation folder of the Microsoft Office Inventory Tool for Updates (the default folder is C:\Program Files\OfficePatch\. You might be required to create this folder).

   - The **Distribution Settings** page allows you to configure the default objects that are created by the installation wizard. These objects include packages, programs, collections, and advertisements that you need to deploy the Microsoft Office Inventory Tool for Updates to your SMS client computers. For more information about these default objects, see Table 6.6, "Software Update Inventory Tool Default Objects," earlier in this chapter.

     On this page you can also specify whether or not you want setup to assign the distribution package to all of the distribution points in your site. If you choose not to have this done, the package is not assigned to any distribution points, and you can use standard package management features of the SMS Administrator console to assign the package to the distribution points of your choice.

     The last part of this wizard page prompts you to assign a name to these objects. You should choose a name that allows you to clearly identify the tool and software update type you are installing, and that will allow you to distinguish this instance of the tool from instances that are installed on other sites in the hierarchy.

     Caution

     Renaming these objects after they are created might cause some parts of the software update inventory process to fail.

   - On the **Database Updates** page, specify the name of an Internet-connected SMS client computer to run the Microsoft Office Inventory Sync Tool for Updates task (the synchronization component). The computer that you name here is the synchronization host, and it requires authenticated Internet access through the firewall. For more information about configuring the synchronization component, or for installation on sites without Internet access, see the "Configure the Synchronization Host" section earlier in this chapter.

The installation wizard places the specified computer into a collection and creates a weekly advertisement to download, install, and distribute updated versions of the synchronization component and database. By default, the advertisement is assigned on a weekly basis within the security context of the user who is currently logged on and running the installation wizard.

If you do not supply a computer name and leave the text field blank, the installation wizard creates only the synchronization component program, but not the collection or advertisement.

- On the **Test Computer** page, specify a test computer to be added to the test collection that the installation wizard will create (the pre-production collection). By default, the test collection is specified as the value of the **Limit to collection** property of the main collection. In most cases you will want to add more computers to this test collection after you complete the installation process. For more information, see the "Task 2: Prepare the Test Environment" section earlier in this chapter.

Create the Necessary Collections, Programs, and Advertisements

If you need customized SMS collections, programs, or advertisements that are different from the ones created automatically with the installer program for the software update inventory tools, you can modify the objects that are created after you run the installer program on the site server.

For example, the advertisements for the scan component and the synchronization component are set by default to be downloaded before running from both a local or remote distribution point. If this behavior is not acceptable in your enterprise, you can change it by editing the **Advanced Client** tab in the **Advertisement Properties** dialog box.

Configure the Synchronization Host

There are two ways to configure the synchronization component:

- Attended mode (default)
- Unattended mode

Configuring the synchronization component to run in attended mode

If you are using authenticated firewalls, the attended mode is the best method to use, because it ensures that the synchronization task has authentication through the firewall.

When you run the installer program for either of the software update inventory tools on your site server, it creates a collection, program, and advertisement for the synchronization component based on the settings you specify in the installation wizard. By default, these objects distribute the synchronization component to the computer you designate to act as the synchronization host, where it runs under the security context of the logged-on user.

If you are using attended mode, the synchronization component requires the following:

- The logged-on user must have access to the Internet through the firewall. If authentication is required, an authenticated browser session must be open on the computer. If this is not the case, the synchronization task does not run.
- HTTP 1.1 must be enabled for the registered browser.

- The logged-on user must have **read/write** permission to the package source folder for the scan component.
- The logged-on user must have access to the package object (if the synchronization component will dynamically update the distribution points).

The attended mode has the following potential drawbacks.

- You (or another administrator with the proper credentials) must be constantly logged on to the synchronization host for the synchronization component to work.
- If you are logged off for an extended period of time (for example, on vacation) there could be a delay of software update compliance and a backlog of newly released software updates on your return.

Configuring the synchronization component to run in unattended mode

In the unattended mode, you can configure the synchronization component to operate in a completely unattended manner, without the need for a logged-on user. To do this, you set up a computer to act as the synchronization host under the security context of a local system account. The account that is used is either the LocalSystem account (for computers running the Advanced Client) or the SMSCliToknLocalAcct& account (for computers running the Legacy Client).

Several potential issues exist with this mode:

- Neither the LocalSystem account nor the SMSCliToknLocalAcct& account have network access extending beyond the local computer account. They therefore require the package source folder to be local.
- The firewall/proxy for the synchronization host must allow anonymous access, or you must specify the user name and password for the synchronization task to use in authenticating through the firewall.
- Neither the LocalSystem account nor the SMSCliToknLocalAcct& account have credentials to the package object, which are required to update distribution points following unattended synchronization.

► To configure the synchronization component for unattended operation

Note

You must have **Modify** permission for the package security object type to modify program properties.

1. During software update inventory tool installation, place the synchronization component on the same computer as the package source folder. The package source folder is the location you specify in the **Select Destination Directory** page of the Security Update Inventory Tool Installer or the Microsoft Office Inventory Tool for Updates Installer.
2. Grant the local Administrators group **read/write** access to this folder.

3. In the SMS Administrator console, navigate to the **Programs** item for the software update inventory tool (Security Update Scan Tool or Microsoft Office Inventory Tool for Updates).

```
Systems Management Serve
 └ Site Database (site code - site name)
  └ Packages
   └ package
    └ Programs
```

4. Right-click the program for the synchronization component, click **Properties**, on the **General** tab, modify the command line as follows:

```
Syncxml.exe /s /unattend /site <site server> /code <site code> /target
<package source> /package <packageID>
```

   - On the **Environment** tab, under **Program can run**, select **Whether or not a user is logged in**.

5. Modify the properties for the package to update distribution points on a schedule. You can configure this by using the **Package Properties Data Source** tab.

6. On the synchronization host, start Internet Explorer and open the **Internet Options** dialog box. On the **Advanced** tab, select **Use HTTP 1.1 through proxy connections**, and then click **OK** to save the changes.

7. Ensure that the firewall/proxy settings for the synchronization host allow anonymous access. If not, use the procedure below to specify an authentication account for the synchronization host to use in authenticating through the firewall.

8. Ensure that the source directory for the scan component package is located on the synchronization host. This is because the SMSCliToknLocalAcct& account does not have permissions to update this directory over the network.

Note

If the synchronization host is also a site server, you can remove the /unattend parameter from the command line for the synchronization component program, and you can skip step 5.

Specifying an authentication account for the synchronization task to use

In some network configurations, anonymous access is not allowed through the firewall or a specific proxy host must be specified in order to connect to the Internet. In these cases you can still enable unattended operation for the synchronization component.

The procedure below creates a registry key that specifies a user account and password with credentials for access through the firewall. Although this registry is created in an encrypted form, it is stored such that only administrators may access the data. When the synchronization task runs, the download process on the synchronization host (PatchDownloader.dll) uses the account you specify when it tries to access the Internet through the firewall.

PatchDownloader.dll is also used by the Distribute Software Updates Wizard to download software update files.

Note

When you use the following procedure, PatchDownloader.dll always uses the specified account to authenticate.

To specify an authentication account for the synchronization host to use

1. Locate the program PatchDownloader.exe in the installation directory of the primary site server or SMS administrator console and run it on the computer that is running the synchronization component.
2. The following command line syntax is used for the program:

```
C:\sms\bin\i386\00000409\PatchDownloader.exe /?
Usage: PatchDownloader /s:<server[:port]> [/u:<username>] [/clean]

Example:

PatchDownloader.exe /s:myserver:80 /u:myaccont
```

Where *username* is the credential of an account with access permissions through the firewall. If *port* is not specified, port 80 is used by default. The program will prompt you for the password. To remove the configuration, use the /clean option.

Important

For security reasons, make sure that the account you specify does not have more security credentials than are necessary to connect through the firewall.

Perform a Test Inventory

You should test the software update inventory tools before you distribute them in your production environment.

By default, the installer program for the software update inventory tools creates two collections for distributing the scan component to client computers: the main collection — called <**tool name> (<site code>)** — and a test collection — called <**tool name> (<site code>) (pre-production).** Also by default, the installer program configures the main collection with membership rules that limit the query used to create it to the test collection. This provides an easy way to test the software update inventory tools prior to deploying them.

The "Task 2: Prepare the Test Environment" section, earlier in this chapter, describes the considerations you should take into account when you are setting up a lab for testing the software update inventory tools. After you finish installing the tools on your site server, you can modify the pre-production collection to include all of the computers in your test environment. Then, you can modify the advertisement for the software update inventory tool you are testing. The schedule you specify can be much more aggressive than the one you will use in production.

The procedure below describes another method for expediting the testing of the software update inventory tools. This method is recommended for a small collection of reference computers only.

Important

Using the expedited program causes a full hardware inventory cycle and can cause serious network and performance issues if it is used in your production environment.

To configure the scan component advertisement to perform an expedited inventory

1. In the console tree, navigate to **Advertisements**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Advertisements
```

2. In the contents pane, right-click the advertisement for the scan component, and then click **Properties**.
3. In the **Advertisement Properties** dialog box, click the **General** tab.
4. In the **Program** list, select the expedited program:

```
toolname (expedited)
```

5. Click **OK**.

   SMS sends the updated program data to the client access points in the site.

Verify the Installation

After you complete the setup process for the software update inventory tools, you perform the following tasks:

- Verify that the package and programs that are necessary to deploy the tools are created. To do this, view the packages and programs in the SMS Administrator console.
- Verify that the collections and advertisements that are necessary for the distribution of the tools are created. To do this, view the collections and advertisements in the SMS Administrator console.
- Verify that the client computers send results.

  To do this, in the SMS Administrator console, go to the appropriate collection containing the test client computer, right-click the collection, select **All Tasks**, and then select **Start Resource Explorer**. In the Resource Explorer, expand **Hardware**, and then click **Software Updates**. View the list of all the inventoried software updates for that client computer.
- Review the log file results to view any errors that occurred during installation. The installation wizard automatically displays this log.

- Ensure that the synchronization component of each software update inventory tool is properly configured on the server. The synchronization component downloads the software update database or catalog from the Internet and makes it available to the clients through SMS distribution points. For more information about configuring this component, see the "Configure the Synchronization Host" section earlier in this chapter.
- Verify that the SMSCliToknLocalAcct& account on the site server computer has firewall authentication access and can download updated catalogs. To do this, grant the SMSCliToknLocalAcct& account access to the package source directory.
- Verify that the advertisement for the synchronization component runs correctly to distribute the updated catalogs to the client computers. To do this, view the status messages for the advertisement and check the file dates on the package source folder files and distribution point folders.
- Verify that the correct SMS distribution points are automatically updated to include the latest catalogs. To do this, view the status messages for the advertisement and check the file dates on the package source folder files and distribution point folders. For more information about viewing status messages, see the "Software Update Status Messages" section later in this chapter.
- If the SMSCliToknLocalAcct& account does not have WMI permissions to the package object, the distribution points require a separate, recurring, scheduled update for the latest catalogs, which you configure and add manually. If this is the case, use the **/unattend** option in the command-line interface for the synchronization component to verify that the distribution points are not updated by the synchronization component since the scheduled update would be in effect. For more information about configuring this component, see the "Configure the Synchronization Host" section earlier in this chapter.

Note

Security bulletin catalog data on the Internet is typically updated on a weekly basis, so the time you select for the synchronization tasks should immediately follow that schedule to ensure that the latest updates catalog is available to your enterprise. In the same manner, the distribution of the latest catalog update to each client computer should be scheduled to follow the catalog synchronization for the distribution points. For more information, see the "Scheduling: Best Practices" section later in this chapter.

Distribute the Software Update Inventory Tools to Client Computers

After you are finished testing the tools and verifying the installation, you can deploy the software update inventory scan tools more broadly by removing the test-limited query from the main collection. To do this, you modify the **Collection Properties** dialog box for the main targeting collection.

To remove the test-limited query

1. In the SMS Administrator console, right-click the collection you want to modify, and then click **Properties**.
2. In the **Collection Properties** dialog box, click the **Membership Rule** tab.
3. In the **Membership rules** box, double-click the query-based rule that you want to modify.
4. In the **Query Rule Properties** dialog box, change the selection from **Limit to collection** to **Not collection limited**.
5. Click **OK**.

Tasks for Authorizing and Distributing Software Updates

To determine which of the installed or applicable security updates are necessary for the client computers in your enterprise, you must evaluate each suggested update and then authorize it for distribution within your enterprise by using the Distribute Software Updates Wizard. This phase of the software update management process consists of several tasks:

1. Prepare the package source folder
2. Plan the software update packages
3. Evaluate and prioritize the usefulness and importance of each software update that is determined to be applicable during the audit
4. Isolate and test the update in your test collection before you authorize it for distribution
5. Create or modify the software update packages, using the Distribute Software Updates Wizard.

   This task involves several steps:

   - Configure software update command-line parameters

   Using the Microsoft Knowledge Base articles available for each update, determine the ideal command-line syntax to use when configuring the software update for installation.

   - Configure Software Updates Installation Agent settings

   In this step you control the amount of user interaction, installation grace period and default action, and other installation parameters for the software update package.
6. Configure advertisement settings for the software update package.
7. Test and verify the software update package deployment

The following sections describe each of these tasks in detail.

Task 1: Prepare the Package Source Folder

The package source folder is the folder that the Distribute Software Updates Wizard uses to store all files that are related to the software updates package you create by using the wizard. This folder is very important for several reasons:

- It contains the definitive, tested versions of the software updates that you authorize for distribution in your enterprise.
- It contains information about security vulnerabilities that are known to exist in your enterprise.

For these reasons, it is important that you protect this folder in the following ways:

1. Set the Access Control List permissions on the folder as follows:
   - Grant **Write** permissions to SMS domain administrators only.
   - Grant **Read** permissions to the security context for the SMS executive on the site server. This is either the SMS Service account or the local computer account, depending on the configuration.
   - Do not grant **Read** permissions to users of lower credentials. In particular, do not grant read permissions on the folder to the Everyone group.
2. Back up the folder according to a regular schedule, as determined by the backup policy for your enterprise. For more information, see Chapter 15, "Backup and Recovery."

Task 2: Plan the Software Update Packages

Before you use the Distribute Software Updates Wizard to distribute software updates in your enterprise, you should decide on the strategy you want to use for creating and maintaining software update packages. By default, the software update management components divide software updates into two types: Security and Office. A single package cannot contain both types of software updates, but beyond that a package can contain as many software updates as you choose to include. Deciding on an effective package deployment strategy will help save time in creating, maintaining, and deploying the packages in your enterprise. You should observe the following general principles when planning software update packages for your enterprise:

- Create the packages at the highest level in the SMS hierarchy from which you want to manage software updates. You can then control package deployment at a more granular level by creating advertisements for the packages at child sites.
- A single package can contain multiple software updates, and these updates can be for multiple operating systems, versions, and client locales. At installation time, the Software Updates Installation Agent determines which software updates are applicable to a given client computer, and installs only those updates.
- You can modify existing packages to add newly authorized software updates, remove authorization for a software update, or change installation options.

- You can minimize the number of software updates you need to distribute, and thus the package size, by keeping your client computers current with the latest service pack. For more information, see the "About Service Packs" section earlier in this chapter.
- The Dynamic Package Configuration feature, new with SMS 2003, allows you to specify multiple programs for a single package, and to attach multiple authorization lists. This means, for example, that you can perform a phased rollout of a newly authorized software update, distributing it first to a test collection, next to a small group of early adopters, and only then to the enterprise at large, all from the same package. Another way that you can use this feature is to create a separate program for servers that specifies no automated system restarts and another program for workstations that requires automated system restarts at installation time. For more information, see the "Configure Installation Agent Advanced Options" section later in this chapter.
- The Distribute Software Updates Wizard only lists a software update for approval and inclusion in a package if the update is requested by at least one client computer. You can avoid this limitation by using a reference computer. For more information, see the "Configure Installation Agent Advanced Options" section later in this chapter.

Table 6.7 lists possible strategies for software update packages:

Table 6.7 Software Update Package Strategies: Benefits and Drawbacks

Package strategy	Detail	Benefit	Drawback
Single package containing all authorized software updates; one package for each software update type	▪ Create a single package for all Security updates and another package for all Office updates. ▪ Modify the package periodically by approving newly released software updates to add to the package.	▪ Less overhead in creating a single package. ▪ Can be useful for organizations with homogeneous environments, such as most clients running the same operating system and service pack.	▪ Cannot easily be used to retire product versions or service pack levels. ▪ Can result in large packages, performance problems (especially for mobile clients over slow links).

(continued)

Table 6.7 Software Update Package Strategies: Benefits and Drawbacks *(continued)*

Package strategy	Detail	Benefit	Drawback
Multiple packages organized by operating system or service pack level	▪ Create a package for each operating system version and service pack level. ▪ Create a corresponding collection for each package.	▪ Easily accommodates retiring product versions or service pack levels. ▪ Smaller packages being distributed to each client. ▪ Accommodates heterogeneous environments with multiple client operating system versions.	▪ More administrative overhead in creating and managing packages. ▪ Need to mirror operating system-based collections in test environment.
Base (rollup) package and weekly or as-needed new updates packages	▪ Administer and maintain the base package that contains all authorized updates for update type. The program is configured not to run when no local distribution point is available. ▪ Weekly or as dictated, the administrator also creates dated packages containing only new software updates. Program properties are set to Download and Execute when no local distribution point is available.	▪ Easily accommodates a phased deployment process. ▪ Minimizes size of packages in most active use. ▪ Maintains single Definitive Software Library package for new resources coming online ▪ Can be efficient way of maintaining mobile clients.	▪ More administrative overhead in creating and managing clients. ▪ Multiple patch packages can lead to multiple system restarts if systems have been offline. ▪ Potential for overloading local software cache on mobile clients.

(continued)

Table 6.7 Software Update Package Strategies: Benefits and Drawbacks *(continued)*

Package strategy	Detail	Benefit	Drawback
Packages organized by criticality of software update	▪ Critical security updates. ▪ Non-critical mandatory updates. ▪ Optional updates.	▪ Recommended by Microsoft Solutions Framework.	▪ Administrative overhead caused by Microsoft not having a listing that contains all Critical Security Updates. ▪ Requires multiple advertisements for same users.

Task 3: Evaluate and Prioritize the Software Updates

To determine which of the applicable security updates you want to authorize for distribution to the client computers in your enterprise, you must first evaluate each requested software update. There are many software updates made available every day, and not all of them will be useful to you or appropriate for the needs of your enterprise.

To do this, assess your risks and read about the latest security update information contained in the white papers and Web sites recommended in the "Software Update Management Guidelines" section earlier in this chapter. This process should also include reviewing all associated documentation for each software update, including that sent with the update and supporting information, which may be found, for example, on TechNet (http://www.microsoft.com/technet).

Some of this information can only be gleaned from testing the software update on a reference computer and noting the behavior in your environment, such as:

- What is the wider effect of a particular software update?
- What did the software update change?
- Can the software update be removed after it has been installed?
- What are the dependencies among different environments?
- How can you ascertain that the software was successful?
- What if the patch overwrites specific customizations?
- What are the possible scenarios for restoring a patched environment?

For guidance in deciding which security updates you should apply to avoid an adverse effect in your particular circumstances and in how rapidly you need to take action on given software updates, see the Microsoft Security Response Center Security Bulletin Severity Rating System at http://www.microsoft.com/technet/security/topics/rating.asp.

Task 4: Isolate and Test the Software Updates

The "Task 2: Prepare the Test Environment" section, earlier in this chapter, describes the process of setting up a test lab for software update management. To test an update, you must authorize the update and distribute it to the test collection containing computers with representative configurations for your enterprise.

The testing objectives are as follows:

- Verify that the update installation command-line syntax and installation behavior is what you expected.
- Verify that the user experience (as configured with the Software Updates Distribution Wizard) is what you expected. If your installation contains both Legacy Clients and Advanced Clients, verify that the behavior is acceptable for each client type.
- Verify that the software update performance is what you expected and that it does not adversely affect the performance of any other enterprise application software.

Task 5: Create the Software Updates Packages

In this task, you use the Distribute Software Updates Wizard to perform the following steps:

- View a list of all installed or applicable software updates that have been reported during the last software update inventory.
- Select the software updates that you want to authorize for distribution to your SMS client computers.
- Create or modify the software update packages that you will use to distribute the software updates.
- Optionally, use the Distribute Software Updates Wizard to automatically download the software update files to the package source directory.
- Configure the installation parameters for each software update in the package.
- Configure the user interaction and installation parameters for the Software Updates Installation Agent to use in applying the package.

Important

Be aware that when you authorize a software update for distribution with the Distribute Software Updates Wizard and save the changes to the package, it is very difficult to undo the action. The authorization data (such as time approved and the fact of the approval) persists in several places in the SMS data store. You can, however, stop an authorized update from being distributed by running the Distribute Software Updates Wizard again to modify the package, and then clearing the check box next to the software update in the authorized updates list. To then uninstall a previously installed software update from client computers, you must create a collection query for client computers with the update installed and use SMS software distribution features to distribute an uninstall program for the software update. For these reasons, it is highly recommended that you evaluate and test each software update thoroughly before you authorize it for distribution to your enterprise.

Run the Distribute Software Updates Wizard

The Distribute Software Updates Wizard is installed by default on the computer where you install the SMS Administrator console.

Before you run the Distribute Software Updates Wizard you must:

- Deploy one or both of the software update inventory tools to your SMS client computers.
- Verify that the software update inventory data that is generated by the software update inventory tools has propagated to the site server.
- Have package creation credentials.
- Have collection and advertisement creation credentials, if you choose to allow the wizard to create these objects (recommended).
- Have Internet access from the computer that is running the wizard, if you choose to have the wizard download the software update source files automatically.

Important

You must administer a software update package from the site on which it was created.

Table 6.8 provides a detailed list of the administrative credentials you should have to run the wizard.

Table 6.8 Required Credentials to Run the Distribute Software Updates Wizard

Class	Credential	Detail
Site	Read	Required to run the wizard
Package	Read, Create, and Distribute	Required to create packages with the wizard

(continued)

Table 6.8 Required Credentials to Run the Distribute Software Updates Wizard *(continued)*

Class	Credential	Detail
Advertisement	Read and Create	Not required if you do not use the wizard to create advertisements
Collections	Read, Create, and Advertise	Not required to create packages; required to advertise packages to a collection

► To run the Distribute Software Updates Wizard

1. In the SMS Administrator console, right-click the **Site Database** node or a collection, package, or resource under the Site Database.
2. On the context menu, select **All Tasks**, and then click **Distribute Software Updates**.

The following sections cover some of the information you must provide when you are completing the wizard. For detailed, page-by-page instructions, see the Help that is available when you click **Help** on the first page of the wizard.

Configure Software Update Command-line Parameters

A software update package typically contains a large number of software updates, many of which might be applicable to a given SMS client computer. For this reason, it is possible that a software update package would require multiple system restarts when the software updates are deployed on client computers. To avoid this problem, you should specify command-line options for each software update that provide for no user interaction, no user input, and no automated computer restarts. You can use the **Software Update Properties** page in the Distribute Software Updates Wizard to view and modify the command-line options for each software update.

After installing the applicable software updates for a package, the Software Updates Installation Agent determines whether the SMS client computer needs to restart based on the restart requirements of the individual software updates in the package. For more information, go to the Microsoft Support Web site at http://support.microsoft.com/.

► To configure command-line installation options for individual software updates

- The **Software Updates Status** page of the Distribute Software Updates Wizard displays the software updates you selected. To view and edit properties such as command-line options, select a software update in the list, and then click **Properties**. Using the controls on the page, you can review the Microsoft Knowledge Base articles available for each update and determine the ideal command-line syntax for unattended installation and managing system restarts.

Important

You must specify the correct command-line parameters for each software update. If you include even an extra space when you enter the command-line parameters it might cause the installation of that software update to fail.

Configure Software Updates Installation Agent Settings

The three **Configure agent settings** pages of the Distribute Software Wizard allow you to specify the settings that the Software Updates Installation Agent uses when it installs the software updates from the current package on client computers. These settings control such variables as the amount of user interaction allowed, the grace period and time-out values, and the automatic system restart behavior.

The settings that you specify on these pages should be determined by:

- The degree of criticality of the software updates in the package.
- The role of the client computers that are the destination of the program you are defining.
- The enforcement requirements of your enterprise or of the SMS client computers in the destination collection for the package.

  The sections that follow provide some overview information about the settings that are exposed in these pages.

Configure time-out periods and grace period

The settings on the second and third **Configure Installation Agent Settings** pages allow you to specify the enforcement time periods to be applied by the Software Updates Installation Agent when the advertisement for the current software updates package runs on SMS client computers. This page allows you to configure three settings related to the time period allowed for the software update installation:

- **Countdown (minutes)**

  This setting specifies the amount of time, if any, the Software Updates Installation Agent waits for a user to respond before it takes action.

  The action taken following the countdown depends on the action that you specify in the **After countdown** setting: automatic installation of the update or postponement of installation.

  This countdown is useful when a software update installation is necessary, but no user is present to provide input. Note, however, that the delays that could be caused by such cases are important, because while the user interface for software update installation is displayed, all other software distribution that is using SMS is blocked for that computer.

- **Maximum run time (minutes)**

  This setting specifies the number of minutes the Software Updates Installation Agent waits before determining that the installation of a software update is not progressing due to an unresponsive computer or other installation problem.

Because software updates can come from a wide range of sources with a wide array of behaviors, it is recommended that you proceed with the installation of an update even if it appears to have become unresponsive. However, if a software update is permitted to remain unresponsive for a long period of time, it could leave the system in a vulnerable and inconsistent state. Therefore, it is necessary to set the time-out value to allow an unresponsive update to be disabled. The default setting is 30 minutes. If you enter a value of zero in this setting, the software update is not given any time to be installed. To avoid this problem, you should provide at least 10 minutes for this time-out value as a recommended minimum.

- **Installation grace period radio buttons**

  These three radio buttons on the third page allow you to specify the grace period, if any, that you want to allow users. Variable installation grace periods allow you to prioritize critical updates and provide a flexible installation schedule for less critical updates. There are three types of grace period settings available:

  - **Require updates to be installed as soon as they are advertised** Use this for high-priority, critical updates. This setting makes update installation mandatory.
  - **Users can postpone updates indefinitely** Use this for low-priority updates. This setting allows users an infinite amount of time to install the updates.
  - **Allow users to postpone installation for**: Use this for intermediate priority updates. This setting allows you to create a customized installation schedule.

  If you select the last option, you can set the basis for the grace period either according to the time the update is detected as applicable to the computer or according to the time it was authorized. The grace period can either be enforced per update, or it can be enforced for an entire package of updates. This allows you to include critical and non-critical updates in the same package.

Configure user interaction

The second **Configure Installation Agent Settings** page contains a number of settings that are used for advanced actions, which are discussed in the "Configure Software Updates Installation Agent Advanced Options" section later in this chapter.

The first check box on this page, determines the amount of user interaction that the Software Updates Installation Agent allows during the process of installing the software updates in the package that you are creating or modifying. It is important to understand these settings and how they interact with the settings on the other pages of the wizard to achieve the end-user experience that you require.

Perform unattended installation of software updates (recommended)

This check box determines whether or not notifications are displayed to the end user when software updates are available for installation or are being installed. Preventing users from being aware of system activity can increase security.

When this box is cleared, end users can receive notifications. The nature of the notifications and the actions that are available to the end user depend on the type of client (Legacy Client or Advanced Client) that is running on the user's computer and the other software update installation settings you specify in the wizard.

When this box is checked, end users are not notified of impending or in-progress software update installations and the software updates are silently installed, subject to the default actions you have defined on this page of the wizard. If the installation requires a system restart, the user interface that appears is the operating system's progress dialog box that indicates that a system restart is being initiated.

Important

If you choose to enable silent installations by keeping this check box checked, you should carefully review the other software update installation settings you have configured, such as installation grace period and restart behavior, to make sure that the end result is the behavior you require. For example, if you check this check box but then specify that the software updates computer restart can be postponed indefinitely, then the software updates in the package are never completely installed if they need a computer restart and the computer is not restated.

Notify users about update activity

This check box on the third page is applicable to the SMS Advanced Client only and enables users of the Advanced Client to receive regular notifications of impending software update installations and to postpone or schedule software update installations locally. The notifications occur every three hours. This setting can be used in conjunction with the **Perform unattended installation of software updates** setting and users of SMS Advanced Client computers will receive only reminders that relate to computer restart activity which you might choose to enforce after a future deadline is reached. In more secure environments, this can provide optimal balance of the productivity needs of the user, and the enforcement needs of the administrator.

To configure software update packages to be installed without user notifications

1. In the second **Configure installation agent settings** page of the Distribute Software Updates Wizard, check the **Perform unattended installation of software updates** check box.
2. If necessary, review the other Software Updates Installation Agent settings you have configured for this package/program, in particular the settings on the second **Configure Agent Settings** page of the wizard. Specifically, you should set the following:
   - Under **Specify the restrictions and advanced settings the installation agent should use to install updates that are in this SMS package**:
   - In the **Countdown (minutes)** box, enter **0**.
   - In the **After countdown list**, select **Perform restart**.
   - On the third page, select the **Require updates to be installed as soon as they are advertised** option.

For urgent updates, you can configure the Software Updates Installation Agent to force a restart even if the user has unsaved data on the desktop.

Caution

This option causes possible data loss on client computers.

To configure forced restarts after software update installations

1. Ensure that the software distribution account that is being used has administrative credentials to the destination SMS client computers.
2. On the first **Configure Software Update Client Agent** page of the Distribute Software Updates Wizard, select **Force client programs to close, and discard any unsaved data**.

Configure the Advertisement

The **Advertise updates** page of the Distribute Software Updates Wizard allows you to create an advertisement for the current package/program and to configure some of the basic advertisement properties, such as advertisement frequency. Note that you must have **Create** credentials for the advertisement object to successfully create an advertisement using this page. In many cases, such as creating advertisements for mobile users, you will want to specify more settings than are available on this page. In such cases, you should either create the advertisement manually or edit the advertisement properties after you finish the wizard.

Note

When you click **Browse** to view a list of available collections on this page, be aware that the displayed list contains all collections, whether or not you have privileges to successfully advertise to that collection.

Notes on Deploying Microsoft Office Updates

When you use the software update management components to manage updates to Microsoft Office applications, be aware that there are several irregularities that make the process for distributing Microsoft Office updates more complex:

- The software update inventory tools can only be used on Microsoft Office applications that are installed in per computer mode, not in per-user mode.
- You must configure at least one Office Administrative Point on your site before you can distribute Microsoft Office updates with the wizard.
- There are two types of Microsoft Office installations: client installations and administrative installations. The same software update file cannot be used to update both types of installations. If a computer that is hosting a client installation of an Office product is ever updated from an administrative installation, that computer must be updated from the administrative update files from then on.

- In an update to an administrative installation, the software update installation files must have access to the product code and installation source files of the original installation share in order for the software update to successfully install on client computers.
- Although most Microsoft Office Update files can be downloaded automatically by using the Distribute Software Updates Wizard, many of them are not ready to deploy without further manual steps. These steps can include decompressing the files and downloading and configuring a special tool, Ohotfix.exe. For more information about using this tool, see the following procedure.

About Ohotfix.exe

Ohotfix.exe is a software program that is designed to help administrators deploy Microsoft Office update files within their organizations. Ohotfix.exe works by reading a series of deployment instructions that are contained in an .ini file, and then using those instructions to apply the software update to the computer. Ohotfix.exe can also check applications on the computer to determine which updates need to be applied, and it can order a group of update files so that an installation is optimized.

To install Microsoft Office Update files by using Ohotfix.exe

1. Download the Ohotfix.exe files from the Microsoft Office Web site at: http://www.microsoft.com/office/ork/xp/journ/ohotfix.htm.
2. Place the downloaded files into the package source folder containing the software updates you want to distribute. The following files are required:

```
Ohotfix.exe
Ohotfix.ini
Ohotfix.dll
```

3. Edit Ohotfix.ini using a text editor such as Microsoft Notepad. Instructions on the settings you must provide in the Ohotfix.ini file are contained within the file itself. In particular, however, make sure you specify the following settings to ensure quiet installation:

```
ShowSuccessDialog=0
OHotfixUILevel=q
MSiUILevel=q
```

4. In the package source folder for each Office update you want to distribute, open a command prompt and extract each Office update file using a command such as the example below:

```
C:\path to update file\MyUpdate.exe /c
/t:C:/path to update file
```

> **Note**
>
> Copy the extracted Office update files to the same folder containing the Exe file for the update, and then delete the Exe file.

5. Run the Distribute Software Updates Wizard again and modify the package containing the Office update files you want to distribute. In the **Software Updates Status** page, select each Office update that you want to distribute, and then click **Properties**.

6. In the dialog box that opens, click **Import** next to the **Program** text box and then select Ohotfix.exe. Click **OK**.

   You will see an error message stating that the binary you selected does not match the binaries suggested for this software update. Click **Yes** to proceed.

7. Click **OK** again to close the **Software Update Properties** dialog box. You will see another error informing you that command-line parameters are not specified for this software update. Click **OK**.

Distributing Updates to Administrative Installations

Microsoft Office applications can be installed in two ways: Administrative installations and client installations. Software updates for administrative installations of Microsoft Office products are distributed and applied differently than software updates to client installations. If a computer that is hosting a client installation of an Office product is ever updated from an administrative installation, that computer must be updated from the administrative update files from then on. You cannot distribute a client update to a computer that is running an administrative installation of an Office product, although you can distribute an administrative update to a computer that is running a client installation.

When the Microsoft Office Inventory Tool for Updates runs on SMS client computers, it can report software updates in one of three status conditions: Installed, Applicable, and AdminApplicable. Software updates that are in the AdminApplicable status apply to administrative installations.

Note

Although the SMS status system reports these three status conditions for updates to Microsoft Office applications, the software update reports do not. You can, however, create a custom report that shows software updates that are in the AdminApplicable status. To learn how to create a custom report, see "Create Custom Software Updates Reports" in the SMS Administrator console Help.

Updating administrative installations of Microsoft Office

If your enterprise contains computers that are running client installations of Microsoft Office in addition to computers that are running administrative installations, you should place each group of computers in its own collection and create a separate software update package to distribute to each.

To distribute administrative updates

1. Create a new collection and give it a membership rule that queries on the following:

```
select *  from  SMS_R_System inner join SMS_G_System_PATCHSTATE on
SMS_G_System_PATCHSTATE.ResourceID = SMS_R_System.ResourceId where
SMS_G_System_PATCHSTATE.Status = "AdminApplicable"
```

2. Using the Distribute Software Updates Wizard, create a separate package that contains only administrative updates. Note that when you authorize these software updates for inclusion in the package, you must manually download the necessary files from the Office download site. To do so, click the link to download the update. On the Web page that opens, search for the instructions on downloading the administrative update.
3. Configure an advertisement for the package and distribute it to the administrative update collection.
4. For the computers that are running client installations, create another collection that excludes any computer with an AdminApplicable status by using a query such as the following:

```
select *  from  SMS_R_System inner join SMS_G_System_PATCHSTATE on
SMS_G_System_PATCHSTATE.ResourceID = SMS_R_System.ResourceId where
SMS_G_System_PATCHSTATE.Status != "AdminApplicable"
```

5. Using the Distribute Software Updates Wizard, create a separate package that contains only client updates.
6. Configure an advertisement for the package and distribute it to the client update collection.

Distributing Updates to Windows Installer Applications

Software updates that are distributed to programs that were installed by using Windows Installer have special requirements that must be met to be successfully installed. To apply a software update to such a program, the Software Updates Installation Agent must have access to the original installation source files. In many enterprises, the paths to these source files are not valid over time.

The Windows Installer Source List Resolution feature, new with SMS 2003, allows you to manage software updates to programs that were installed using Windows Installer by ensuring that the original installation files are always available to the SMS client. To use this feature, you must first enable it by changing the program's package properties using the procedure below.

► To specify a source file location for a Windows Installer package

1. In the SMS Administrator console, navigate to **Programs**:

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Packages
   └ package
    └ Programs
```

2. In the details pane, right-click the program that you want to modify, and then click **Properties**.
3. In the **Program Properties** dialog box, follow the instructions on the **Windows Installer** tab to provide the source location for the package.

After you have specified the source file location for the program package, you can authorize software updates for distribution to SMS client computers that are running that program. When you authorize a software update to a Windows Installer program by using the Distribute Software Updates Wizard, you can now specify file names in the Windows Installer file format (.msi or .msp). Using the Distribute Software Updates Wizard, you can create or modify the package that you want to contain the software updates. To do so, use the following procedure.

► To specify Windows Installer files in the Distribute Software Updates Wizard

1. On the **Add and Remove Updates** page, select the software update that you want to authorize.
2. On the **Software Updates Status** page, select the software update, and then click **Properties**.
3. On the **Software Update Properties** page, click **Download** and perform the steps to download the software update files, and then manually decompress the files. For information about how to do this, see the "Notes on Deploying Microsoft Office Updates" section earlier in this chapter.
4. On the **Software Update Properties** page, type the name of the Windows Installer file (.msi or .msp) in the **Program** box or click **Import** to browse to the file in the package source folder.
5. In the **Parameters** box, specify the command-line options that the Software Updates Installation Agent must use when processing the software update on SMS client computers. Note that .msi and .msp files are automatically processed using the Msiexec.exe command, so the command-line options you supply here should be the options for that command. For example, to specify that the software update is installed without user interaction and with automatic restart suppressed, you would specify the following:

   ```
   /q REBOOT="ReallySuppress"
   ```

   Note, however, that when the command runs on the client, the actual command-line that the Software Update Installation uses in this case would be:

   ```
   msiexec.exe /i <patch.msp> /q REBOOT="ReallySuppress"
   ```

   Where <patch.msp> is the Windows Installer file you specify in the **Program** box.

For more information about Windows Installer command-line options, see MSDN at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/setup/command_line_options.asp.

Configure Software Updates Installation Agent Advanced Options

The Distribute Software Updates Wizard and the Software Updates Installation Agent have advanced configuration options. The following sections describe these options and give procedures for using them.

Use a reference computer to expedite approval processing

Because the Distribute Software Updates Wizard does not list a software update for approval until the update has been requested by at least one client computer, there might be some delay between the time a software update becomes available and the time it is approved for distribution. To minimize this delay, you can use this procedure to bypass the collection-wide software inventory process and add the software update to the software updates authorization list based on the inventory of a single reference computer. This is useful when critical software updates must be distributed immediately. The following procedure describes how to create a reference computer template. To learn how to import the template that you create into the package or program that you want to distribute, see the "Specify a New Software Updates Authorization List" section later in this chapter.

To create a reference computer template

1. Configure an SMS client computer so that it represents the production environment of the target computers for the package/program that you want to distribute. This is the reference computer.

   Note

   For ease of deployment and tracking, place the reference computer in its own collection.

2. If you have not already done so, deploy the software update inventory scan component to the reference computer. Make sure that the latest version of the software updates catalog is available (for example, Mssecure.cab).

   Note

   You can download the file manually. For example, for the Security Update Inventory Tool, download the file at http://www.microsoft.com/smserver/downloads and then copy it to the installation folder of the Security Update Inventory Tool (the default folder is C:\Program Files\Security Update\1033.)

3. Run the Distribute Software Updates Wizard to either modify an existing package or to create a new package. The content of this package is unimportant; you are only using it to force the Software Updates Installation Agent to output the local version of PatchAuthorize.xml that you will use as a reference template. Make sure, however, that the package is of the same software update type as the software updates that you are concerned with.

4. Step through the wizard to configure the package. Make sure you specify the following items:
   - You must select at least one software update for authorization to complete the wizard.
   - On the last **Configure Installation Agent Settings** page, select the **Create reference computer templates during processing** check box.
   - On the **Advertise updates** page, select the **Advertise** check box. Under **Collection**, browse to the collection that contains your reference computer.
5. After you complete the wizard, right-click the advertisement that was created for the new package, point to **All tasks**, and then click **Re-run advertisement** .

   When the advertisement runs, the Software Updates Installation Agent creates a file called `<type>_PatchAuthorize.xml` (where *type* is the software update type) in the system temp folder of the reference computer where you ran the advertisement (for example, C:\winnt\system32\temp). This file contains a master list of all the software updates that are detected on the reference computer, whether installed, applicable, or authorized.
6. You can import this new authorization list into a new or existing software updates package to distribute software updates to SMS client computers in your production environment based on this authorization list. To learn how to do this, see the "Specify a New Software Updates Authorization List" section later in this chapter.

Configure scheduled software update installations

Using the advanced configuration options in the Distribute Software Updates Wizard, you can schedule software update installations to begin and end at a specific time. This is especially useful in unattended installation scenarios such as server updates, where installation of software updates and required restarts must not happen outside a certain time period. If a scheduled installation is configured and installation does not occur within that time period, the software update installation is postponed until the next occurrence of the specified time period.

Important

Be careful when you use this feature with the persistent notification feature. For example, it is possible that notifications will appear outside of the scheduled time period when installations are actually allowed, leading to potential end-user confusion. In general, scheduled installations are designed to be used in silent installations that require no user interaction. For more information, see the "Configure user interaction" section earlier in this chapter.

To configure a package/program for scheduled installation

1. Run the Distribute Software Updates Wizard and create or modify the package containing the software updates that you want to assign for scheduled installation.
2. On the second **Configure Agent Settings** page, select **Use a restricted installation start time and duration when processing updates and permitted system restarts**.

3. In **Wait <N> minutes maximum for all updates and then defer remaining items** type the number of minutes you want to allow for the software update installation after the advertisement begins to run.
4. Step through the rest of the wizard, and then click **Finish** on the last page.
5. Follow the steps to create an advertisement for the package you just created or modified. On the **Schedule** tab in the **Advertisement Properties** dialog box, under **Advertisement Start time**, specify the start time for the scheduled software update installation. The start time you specify will be the time that the scheduled installation begins.

Enable dynamic package configuration

Dynamic package configuration is a powerful new feature for advanced users of the software update management components. With dynamic package configuration, you can configure multiple program objects for the same package. Each program object can have its own properties. This allows you to perform such tasks as:

- Differentially distribute the same package to multiple collections with different installation options for each collection.
- Add a new software update to a package and distribute it to a test collection first, before authorizing it for distribution to the rest of your SMS client computers
- Perform progressive installations of a software update package to successive groups of SMS client computers, each targeted with a program set to a specific scheduled installation time period.

To use the dynamic package configuration feature, first run the Distribute Software Updates Wizard in the usual way to create the default program for the package. Then use the procedure below to create a second program. You can create as many programs as you want for a given package.

► To specify a new program object for an existing package

1. Run the Distribute Software Updates Wizard to create a software updates package or modify an existing package.
2. On the Identify the SMS package page, click Advanced.

   The **Program Item Settings** page appears and displays the name of the current program. Unless you have previously created a dynamic package, this will be the default program with the name of the package.
3. Click **New** to create a new program object for the package.
4. In the **Program name** box, type a name for the new program.
5. Optionally, attach a new software updates authorization list to the new program or merge the contents of an existing authorization list. For more information, see the following section.
6. 9Click **OK**.

After you create the new program object, any settings you then configure with the wizard apply to that program. Any software updates that you authorize are added to the package but are approved for authorization for that program only. You can also use the wizard to configure an advertisement for that program, and assign the advertisement to the collection of your choice.

For example, you can use a reference computer template to generate a new authorization list that lists a software vulnerability that has not yet been reported by client computers in your enterprise. You can use the procedure in the following section to attach the new authorization list to the program, authorize the new security update for the vulnerability, and then create an advertisement and assign the advertisement to your test collection.

Specify a new software updates authorization list

As described in the "How Software Update Management Works" section earlier in this chapter, the Software Updates Installation Agent uses a software updates authorization list to determine which software updates to install on SMS client computers. The Distribute Software Updates Wizard creates the default version of this list (PatchAuthorize.xml) when you originally run the wizard to create a package.

You can specify a different authorization list for a package or program that you create with the wizard. You might want to do this, for example, if you need to authorize a software update that is newly released and has not yet been reported as missing on any client computer.

There are two ways to specify a new authorization list for a package.

► To attach or merge another software updates authorization list to a package or program

1. Generate the new software updates authorization list that you want to attach. For example, you can use the procedure defined at the beginning of this section to create a reference computer template.
2. If necessary, copy the file you created in step 1 to the package source folder containing the software updates package you want to update.
3. Run the Distribute Software Updates Wizard to create a software updates package or modify an existing package.
4. On the Identify the SMS package page, click Advanced.
5. The **Program Item Settings** page appears and displays the name of the current program and the authorization list that is attached to that program.

   Unless you have previously created a dynamic package, **Program name** is the default program with the same name as the package, and **Authorization list** has the default file name of PatchAuthorize.xml.
6. Select the program to which you want to attach the new authorization list, or click **New** to create a new program.
7. In the **Authorization List** box, type the name of the new authorization list file that you created in step 1.

   – Or –

   Under **Authorization List**, click **Import**. In the **Windows file chooser** dialog box, navigate to the authorization list you want to merge, and then click **OK**.

Important

When you merge a software updates authorization list, items in the newly merged list take precedence over duplicate items in the existing list.

To create a new software updates authorization list

1. Run the Distribute Software Updates Wizard to create a software updates package or modify an existing package.
2. On the Identify the SMS package page, click Advanced.
3. The **Program Item Settings** page appears and display the name of the current program and the authorization list that is attached to that program.

   Unless you have previously created a dynamic package, **Program name** is the default program with the same name as the package, and **Authorization list** has the default file name of PatchAuthorize.xml.
4. Select the program to which you want to attach the new authorization list, or click **New** to create a new program.
5. Click **OK** to close the **Program Item Settings** box and return to the wizard.
6. Click **Next**.

   A message appears warning you that the file does not exist and asking if you want the wizard to create it for you.
7. Click **OK**.

Task 6: Customize the Package and Advertisement Settings

The following are points to consider when configuring the advertisement settings for a software updates package

- Advertise first to a test collection of systems in your controlled lab environment. When each system has been verified, you can proceed to a broader target group, such as a production pilot group.
- Set the recurrence feature to a value that allows end users to have several opportunities to become involved in the process, but not so often as to be annoying to them or cause undue disruption.
- Consider the enforcement period when setting the recurrence value. For the example of a seven day enforcement period with a 6 hour recurrence, end users will have 4 recurrences per day or 24 opportunities a week, but typically only 10 of these will occur during usual business hours.

- Also consider that Advanced Clients have the option of the persistent notification feature, which provides a local reminder at three-hour intervals, independent of the advertisement schedule. You should therefore configure the advertisement schedule based on the number of Legacy Clients in your environment and the need to simulate a reminder-like behavior for those clients.

Task 7: Test the Software Update Packages

To ensure that patches are tested, and that Security Patches are recognized as quickly as possible, do the following prior to going into production and prior to deploying security patches. This requires a permanent lab, but it can be connected to the rest of the network and does not have to be isolated from the production LAN or domains:

If you have a lab, include reference computers that represent one of each Microsoft operating system and version that you use in production. These systems should be as identical as possible to what you are running in your production environment, including service packs, hardware, operating systems, applications and antivirus software.

- Verify notification behavior.

  If your client computers are running Windows 2000 or later, verify that the notifications (balloons) that indicate software update installation processes function as expected. Note that computers running Windows NT 4.0 operating systems do not display notification balloons, but rather, display a notification icon in the system tray and display dialog boxes.

- Verify the grace period.

  Ensure that the grace period for software update installation is enforced.

  To do this, set a grace period for update installation by using the **Configure Installation Agent Settings** page in the Distribute Software Updates Wizard or the command-line interface for the agent. Allow the grace period to expire, and then verify that the update installs automatically.

  Note that when the persistent notification feature is enabled on the Advanced Client, the grace period is observed independently from the advertisement schedule. For example, if the advertisement will not run for another five days, but the grace period for an update will be reached in two days, a local copy of the advertisement will run on the client in two days.

- For packages with multiple updates, verify that grace period enforcement is based on the time the oldest applicable update in the package was authorized.

  To do this, create a package that contains multiple updates with different authorization dates (you can configure the authorization date for an update by clicking **Properties** in the Distribute Software Updates Wizard). Set the grace period for the entire package. Verify that the grace-period expiration time is correct, based upon the oldest authorization date.

- Verify that the per-update grace period enforcement leaves unexpired patches in an optional state.

  To do this, create a package that contains multiple updates, and configure per-update grace period enforcement by using the **Configure Installation Agent Settings** page in the Distribute Software Updates Wizard. Allow the grace period to expire, and then verify that the only updates that have mandatory installation status are those whose grace period has expired.

  The non-expired updates should be available for installation, but not mandatory; they are installed only if the user clicks **Install Now**. If the countdown timer reaches zero and the agent initiates the installation process, the updates for which the installation grace period has not expired are not be installed automatically.

- Verify default action.

  Ensure the specified failsafe time-out, installation countdown, postponement and default installation actions occur properly if no user interaction is provided.

  To configure these settings, use the Distribute Software Updates Wizard or the Software Updates Installation Agent command-line syntax.

  Both SMS and the Feature Pack tools support notification and countdown features for assigned programs. When using the Feature Pack tools to deploy software updates, it is recommended that you disable the SMS versions of the countdown and notification features to prevent confusion. If the SMS versions of these features remain active, end users see two sets of countdowns and two sets of notifications for each assigned program.

- Verify branding.

  To test whether your branding is appearing properly, create a file, named Summary.htm, in the package source folder, and place some branded content in it. Then, verify that your client computer properly displays the branding. Note that embedded objects such as graphics do not appear on computers that are running Windows NT 4.0.

  Branding is specific to each package, so when you configure branding for a package all updates in the package share the branding. Different packages can have different branding, for example, Critical Updates in one package, and Office Updates in another package, each with different branding.

- Verify failsafe time-out behavior.

  Test the failsafe time-out behavior by using the **Parameters** field and clicking **Syntax** on the wizard properties page to configure an update that does not suppress user input (that is, it requires user input to install) and then verify that the update is terminated after the time-out has been reached.

  Also, after that update terminates, verify that the Software Updates Installation Agent attempts to install the remaining updates in the package.

- Examine status data.

  Verify whether the status data for updates is accurate by checking to see if the TimeApplied value is correct for all installed updates processed by the Software Updates Installation Agent. This information can be viewed in the inventory schema found within the SQL View: v_GS_PATCHSTATE, from the SMS Resource Explorer or from the sample reports included with the Reporting add-in.

- Verify system restart behavior.

  You can configure system restart behavior by using the **Configure Installation Agent Settings** page in the Distribute Software Updates Wizard or the Software Updates Installation Agent command-line interface.

  You can configure different post-installation system restart behavior for workstations and servers in your enterprise. Based on the settings you configured, ensure that restart detection will function as you expect for each computer role. To do this, configure different system restart settings for different updates, and then monitor the behavior of the system installing the update.

  When a system restart is required, the closure of active applications can be configured with a countdown to restart. This provides users with the opportunity to save their work. Alternatively, applications can be closed and the system can be restarted without a grace period. Verify that application closure during post-installation system restart will function as you expect.

Task 8: Expedite Delivery of New or Urgent Updates (optional)

Occasionally, you might need to deploy a software update very rapidly, such as during an attack of a newly released virus or worm. Because the software updates that address such threats are often available long before the threat becomes active, it is common for the item to be listed in the Distribute Software Updates Wizard interface for pre-authorization. After you authorize the software update, you can quickly deploy it into your testing and production environments by using the steps described in this section. This is an optional task, and you should reserve it for urgent cases because these steps might temporarily reduce network and system performance.

Use the following guidelines for preparing your environment to enable expedited delivery of new or urgent updates:

- Clients process new advertisements according to their polling interval settings. For this reason, you should set the client polling interval for the Advertised Program Client Agent to values that are appropriate for both your expected response time during urgent cases and the network and server load that is acceptable during non-urgent cases.

  To do this, use the following procedure:

► To set the client polling interval

1. On the **General** tab in the **Advertised Program Client Agent** dialog box, configure the **program polling interval** (for the Legacy Client) and the **policy polling interval** (for the Advance Client).
2. The delta replication feature in SMS 2003 allows you to distribute the changed authorization list and added files for the software update much faster than with SMS 2.0. Depending on the network settings for your site-to-site communications, however, there might be some delay in how quickly the changes to the package can replicate to child sites and clients. To prevent this, ensure that your intersite bandwidth settings are consistent with the advertisement and package sending priority you usually use, so that you always have the option of setting the priority to High for an urgent new update and thus can bypass the bandwidth restrictions in those urgent cases.

The following procedure describes a method for initiating a one-time forced re-run of a software update package advertisement prior to the next recurrence date for the advertisement. Clients process new advertisements according to their polling interval settings. For this reason, you might choose to use a new package or a new program to expedite the delivery of an urgent update. Existing advertisements observe their recurrence schedule (weekly by default) and are the primary deployment method for normal operations.

► To expedite delivery of a new or urgent update

1. Using the Distribute Software Updates Wizard, create or modify a package to contain the software update you want to expedite.
2. Complete the authorization of the software update by using the appropriate enforcement settings (consider setting the authorization date to a past date to ensure that the software update becomes required sooner than the usual grace period would allow).
3. In the SMS Administrator console, open **Advertisements**, and then right-click the advertisement associated with the program you configured with the Distribute Software Updates Wizard in step 1.
4. On the context menu, select **All Tasks**, and then click **Re-run Advertisement**.

   For more information about this feature, see Chapter 5, “Distributing Software.”

This procedure forces the advertisement to run on all clients in the collection to which the advertisement is assigned, and causes the new software update to be installed on clients where the update is applicable, subject to the enforcement settings you specified for the package/program.

Monitoring Software Update Distributions

SMS 2003 provides several features that allow you to track and evaluate software update inventory, requirements, installation, and compliance within your enterprise. You can use these tools to spot problem areas quickly and easily.

You can use the same tools that you use to monitor software distribution to monitor the progress of a software update distribution in your enterprise. These tools, such as the **Package Status** summary and the **Advertisement Status** Summary, are described in Chapter 5, "Distributing Software." In addition to these tools, SMS 2003 provides a number of tools and features that are specific to software update management. These tools and features are described in the following section.

Tools for Monitoring Software Update Distributions

At various points in the software update management process, you can use SMS tools to report compliance levels for specific vulnerabilities, monitor the status of software update distributions, check the health of the software update management components, and troubleshoot software update compliance. For example, if a new critical update is released for a particular vulnerability in Windows 2000, you can run a report that shows all computers that are running Windows 2000 in your enterprise that are missing that critical update. When you authorize and distribute that software update, you can periodically run another report that shows compliance levels as reflected in hardware inventory and status messages.

Table 6.9 lists the features that are available for monitoring software update processes.

Table 6.9 Monitoring Features for Software Update Management

Feature	Description
Software update status messages	Software update status reporting provides real-time information about the installation progress of specific software updates on specific computers. Several of the SMS reports for Software Update Management draw on the software update status system for current information about the progress of a deployment.
Software update compliance reports	Software update reports are available from the SMS report viewer and include information about software updates or client computers, such as update detection time and update installation time. This information allows you to track the progress of a specific update or to check the update status for a specific computer.
Software update distribution status reports	These reports help you evaluate the effectiveness of your software update management practices and assess the areas of risk in your enterprise.
Software update infrastructure health reports	These reports help you monitor the performance of your software update management components and troubleshoot failed software update installations.

(continued)

Table 6.9 Monitoring Features for Software Update Management *(continued)*

Feature	Description
Custom reporting from a rich, documented schema	The Software Updates category of SMS 2003 reporting contains several pre-configured reports that you can use to view software update specific information. In addition to using the preconfigured reports, you can also use SQL Server views and the documented inventory schema to create custom software update inventory reports, tailored to the needs of your enterprise.

These tools are described in the following sections.

Software Update Reporting

To understand the information in this section, see Chapter 11 "Creating Reports."

A variety of predefined reports are provided with SMS 2003 to help you quickly obtain information about the software update status of your enterprise. These reports are designed to provide views of current compliance levels and distribution status and to provide data to support trend analysis and troubleshooting.

The software update management reports can be found in the **Reports** item of the SMS Administrator console under the following categories:

- Software update — compliance
- Software update — distribution status
- Software update — infrastructure health

The following sections discuss each of these categories in detail.

Software Update Compliance Reports

These reports use a combination of software update inventory data and software update status summarizer data to provide a near real-time snapshot of the software update compliance level in the enterprise. Reports in this category cover compliance for specific software updates or for a specific product, in addition to providing various views on the overall compliance status of the enterprise. This information is useful for managers who need to assess exposure to specific vulnerabilities for which a software update has been released and for planning the scope and phasing of a software update deployment.

Software Update Distribution Status Reports

These reports address the distribution status of software updates that have already been authorized and distributed in the enterprise. Reports in this category cover the installation status of specific software updates or all authorized updates, in addition to providing data on the number of computers that display a specified software update installation status. This information is useful for monitoring the progress of a software update distribution and for troubleshooting unsuccessful deployments.

Software Update Infrastructure Health Reports

These reports provide information about the health of the SMS software update management infrastructure, such as software update management components that are reporting error status and SMS client computers where software updates cannot be installed. This information allows system administrators to troubleshoot software update distribution problems and monitor the reliability of their software update management processes.

Software Update Status Messages

Several of the software update management client and server components generate status messages that you can use for troubleshooting and for determining the status of a software update distribution. Additionally, you can use the SMS status messages that are generated by other SMS components (such as packages and advertisements) to gain a complete picture of your software update management components and processes.

To understand the information in this section, see Chapter 14 "Using the SMS Status System."

Software Update Management Component Names

Both client and server components of the software update management system generate status messages. You can view these status messages directly, by constructing a status message query, or you can view the output of these messages in various predefined reports. In addition to the software update reports, for example, you can use the reports in the **Status Messages** and **Status Messages – Audit** category to quickly and easily access the status messages by component, client, or error level.

Table 6.10 lists the software update management status components and describes the messages they produce.

Table 6.10 Software Update Management Components in the SMS Status System

Component	Description
Distribute Software Updates Wizard	Sends audit status messages when new software updates are authorized.
Software Updates Installation Agent	Reports events related to software update installation on client computers. Provides information about installation status that is used by many of the software update reports. For a list of possible software update installation status conditions reported by this component, see Table 6.12.
Software update scan component	Reports events related to software update inventory scan process on client computers. Note that this component name does not distinguish which software update inventory tool is in use, although the specific software update type is specified in the body of the message.

(continued)

Table 6.10 Software Update Management Components in the SMS Status System *(continued)*

Component	Description
Software update synchronization component	Reports events and errors related to the software update inventory synchronization component. Note that this component name does not distinguish which software update inventory tool is in use, although the specific software update type is specified in the body of the message.

Software Update Logging

All of the software update management client and server components maintain log files

The Software Updates Installation Agent maintains a log file on each SMS client computer. You can look at this file to determine the status of software update installations. You can also look at the log files that are maintained by the individual software updates as they are installed.

Table 6.11 lists the software update installation log files and their locations.

Table 6.11 Software Update Installation Client Log Files and Locations

Component	File name	Location	Description
Security Updates Sync Tool (Syncxml.exe)	SecuritySyncXml.log	Synchronization host, in the Temp folder of the account running the process (current user if running in attended mode; system temp if running in unattended mode).	Log file for the synchronization component; used for troubleshooting firewall and authentication issues.
Microsoft Office Inventory Sync Tool for Updates (Syncxml.exe)	OfficeSyncXml.log	Synchronization host, in the Temp folder of the account running the process (current user if running in attended mode; system temp if running in unattended mode).	Log file for the synchronization component; used for troubleshooting firewall and authentication issues.
Security Updates Scan Tool (S_scan.exe)	SecurityPatch.log	System temp folder on SMS client computer.	Log file maintained by scan component on SMS client computer.
Microsoft Office Inventory Scan Tool for Updates (O_scan.exe)	OfficePatch.log	System temp folder on SMS client computer.	Log file maintained by scan component on SMS client computer.

(continued)

Table 6.11 Software Update Installation Client Log Files and Locations *(continued)*

Component	File name	Location	Description
Software Updates Installation Agent	PatchInstall.log	System Temp folder of the SMS client computer.	Package installation log file maintained by the Software Updates Installation Agent on the SMS client computer.
Individual software update files	<qnumber>.log	%*Windir*% folder on SMS client computer.	Installation log maintained by software update installers. Contains information about actual software update installation.

Tasks for Monitoring Software Update Processes

To determine whether your software update deployment is successful, you can use SMS software update management components to track the progress of software update compliance in your enterprise. Monitoring tasks include:

- **Auditing the Enterprise for Current Security Vulnerabilities** Determine which software updates are missing and applicable in your enterprise or on a particular computer or software version.
- **Monitoring the status of software update distributions** Find out the progress of software updates that you have already authorized for distribution in your enterprise.
- **Checking the health of software update management components** Detect problems in scan component functioning, synchronization component download or authentication errors, and other software update management components.
- **Troubleshooting software update installation errors** Spot problems, trends, or errors in your software update management process.

Task 1: Audit the Enterprise for Current Security Vulnerabilities

When new software updates are released to address recently identified security vulnerabilities, it is often necessary to conduct an enterprise-wide audit of the breadth and depth of exposure to the vulnerability to determine a strategy for successfully addressing it. Current status information is required for such an audit to be successful. This status information is available through a combination of tracking mechanisms.

Auditing with SMS Software Update Reporting

The SMS reports in the **Software update – compliance** category provide several views into the current compliance status of your enterprise.

These reports can help you obtain such information as:

- **Service coverage** — How many systems are currently in compliance for the software update.
- **Impact** — How many systems require the software update.
- **Exposure** — How many systems are currently out of compliance for the update.

Auditing with Other SMS Features

When a new, critical software update is released, you can also use SMS hardware and software inventory to query clients according to criteria in the vulnerability matrix for update. This is not necessary for deploying the software update, but it can be useful for determining the overall exposure to the vulnerability, and whether or how aggressively the software update should be deployed.

For example, if the vulnerability only exists on computers that are running Internet Information Services, and no computers in a collection are running Internet Information Services (IIS), the software update deployment can be skipped for that collection.

Task 2: Monitor the Status of Software Update Distributions

When you authorize software updates for distribution in your enterprise, you should monitor the progress of the distribution among the SMS client computers that are targeted to receive those software updates.

Monitoring with SMS Software Update Reporting

The SMS reports in the **Software update — distribution status** category are designed to help you confirm the coverage being achieved for software updates that you have already deployed in your enterprise, and to identify client computers that are returning a failure status for those updates. This allows administrators to identify common criteria for computers that are failing.

These reports query a combination of inventory data and per update and summary status messages to give a snapshot of the current compliance level that is close to real time.

These reports display information such as:

- The number of computers that are reporting a particular software update distribution status (such as failure and success).
- The distribution progress of a particular software update.
- A summary of the distribution status of all authorized software updates that have been deployed to a particular collection.

Many of these reports list the distribution status of each specific software update. The distribution status property is an optional property of software update status messages, and indicates the current status of the installation of a specific software update on a specific client computer. Table 6.12 shows the distribution status categories and their meanings.

Note

The software update reports use slightly different terminology than software update status messages when referring to distribution status.

Table 6.12 Software Update Installation Status

Distribution status	Description
Success (This status is also called **Install verified** or **Distribution successful** in software update reports.)	The software update installed successfully and a restart was either not required or was successfully performed. For specifics, see the message.
Restart pending	The software update installed successfully and a system restart was required but has not yet been performed, because the restart was either automatically postponed or postponed by the user. For specifics, see the message.
Retrying	The software update installation was attempted but was unsuccessful for one of a variety of non-failure reasons. For details, see the specific message. The installation will be attempted again the next time the advertisement runs.
Postponed	The software update installation was postponed either automatically or by the user. For details, see the specific message.
Failed	The software update installation failed due to an error condition. For possible reasons, see the specific message.
Uninstalled	A previously installed software update was uninstalled by the user or by another process independent of the software update management components.
No status (reports only)	No status messages have been received for the specified software update.
Distribution incomplete (reports only)	A general reporting category that combines the distribution status categories of **Retrying**, **Restart pending**, and **Postponed**.

Monitoring Distributions with Other SMS Features

You can also determine the status of a software update distribution to an SMS client computer by viewing the software update inventory data for that computer in Resource Explorer. The **status** category for that software update changes from **Applicable** to **Installed** when a software update has been successfully installed on the client computer.

Note

Software updates for Microsoft Office applications can have a third status in Resource Explorer, AdminApplicable. This status applies to software updates to client installations that are being managed from an administrative shared folder. For more information, see the "Notes on Deploying Microsoft Office Updates" section earlier in this chapter.

Be aware, however, that the information displayed in Resource Explorer is only as accurate as the most recent hardware inventory data.

Task 3: Check the Health of Software Update Management Components

Another important task related to monitoring software update processes is monitoring the successful performance of the tools and components related to software update management. This task should be performed regularly according to the needs of your enterprise.

Monitoring Infrastructure Health with SMS Software Update Reporting

The SMS reports in the **Software Update — Infrastructure Health** category are designed to help you monitor the performance of your software update management components and processes by reporting such data as:

- Client computers that are generating software update installation error messages.
- Runtime or download errors being generated by the scan component, the synchronization component, or the Software Updates Installation Agent.

Monitoring Infrastructure Health with Other SMS Features

Use the Advertisement status summarizer in the SMS Administrator console to determine the success or failure of the advertisements you created for the following:

- Software update packages
- Software update inventory tool scan component
- Software update inventory tool synchronization component

Task 4: Troubleshoot Software Update Installation Errors

You perform this task on an as-needed basis to identify software update installation failures or exceptions and then track down and resolve the causes. Exceptions typically follow a pattern that can be resolved by refining your software update management process.

Troubleshooting tasks include:

- Spotting trends (for example, the software update compliance level is not increasing).
- Narrowing issues (for example, status messages indicate failures).
- Determining problems (for example, the software update you downloaded is for the wrong operating system).

For example:

- If inventory reports are run daily, but inventory schedules occur on a weekly or monthly basis, the reports that you view might not indicate that progress has occurred until the scheduled inventory happens.
- There might be fewer computers than expected in the targeted collection, and a review of the collection rule query might be necessary.

Troubleshooting with SMS Software Update Reporting

The SMS reports in the **Software Update — Distribution** category and the **Software Update — Infrastructure Health** category can be useful to help troubleshoot installation errors. These reports can help you determine:

- Which client computers are reporting errors for a specified software update.
- Which client computers are in a specified error condition.

Troubleshooting with Other SMS Features

In addition to viewing software update reports, you can view software update status messages and software update log files to help give more specific information about the reasons of a software update installation failure. For example, if a software update installation was attempted but could not be completed before time-out occurred, the information about this error is likely to be contained in the log file that is maintained on the client computer by the software update installation program itself. For more information, see the “Software Update Logging” section earlier in this chapter.

There are also several Knowledge Base articles, available at http://support.microsoft.com, that can assist you with the process of fine-tuning your software update management process by providing information about how to troubleshoot inventory, software distribution, and status message processing.

Software Update Management Best Practices

This section briefly describes recommended best practices for managing software updates to help administrators avoid common problems.

General Best Practices

The best practices listed in this section are described in more detail in the *Patch Management Using SMS/Deployment Guide* white paper, which is available at the Microsoft Solutions for Management Web site at http://www.microsoft.com/solutions/msm.

Perform an initial audit

An audit helps an organization understand and gain an accurate record of its technology assets, prior to initiating a software update management program. Accurate and current information of what is present in the production environment is essential for software update management.

Establish baselines

An important part of the software update management process is creating initial standard installations of operating system versions, applications, and hardware for computers in your enterprise, called *baselines*. A baseline is the configuration of a product or system established at a specific point in time. An application or software baseline, for example, provides the ability to rebuild a computer to a specific state.

Baselines provide the basis for finding and fixing potential problems and simplifying the software update management process considerably, both by reducing the number of software updates you must deploy in your enterprise and by increasing your ability to monitor compliance.

After performing the initial audit of your enterprise, you should use the information that is obtained from the audit to define an operational baseline for the IT components within your production environment. A number of baselines might be required, depending on the different types of hardware and software deployed into production. For example, certain laptop computers require a software update to prevent them from hanging when they enter hibernation or standby mode when running Windows XP. A baseline for these laptops should include this software update.

In large organizations, it is often helpful to divide the computers in your enterprise into asset categories and keep each category at a standard baseline by using the same versions of software and software updates. You can then use these asset categories in prioritizing a software update distribution.

The Software Updates Installation Agent includes an option to generate a reference computer template that contains the baseline of software updates from a reference computer. For more information, see the "Use a reference computer to expedite approval processing" section earlier in this chapter.

Subscribe to the appropriate software update notification services

After you perform an initial audit of the software in use in your enterprise, you should determine the best method for receiving notifications of new software updates for each software product and version. Depending on the software product, the best notification method might be e-mail notifications, Web sites, or computer publications.

For example, the Microsoft Security Response Center (MSRC) responds to all security-related concerns about Microsoft products and provides the Microsoft Security Bulletin Service, a free e-mail notification of newly identified vulnerabilities and software updates that are released to address these vulnerabilities. You can subscribe to this service at http://www.microsoft.com/technet/security/bulletin/notify.asp

Note that when receiving e-mail notifications for software updates, you should always verify the validity of the message. For more information, see the *Patch Management Using SMS/Deployment Guide* white paper at http://www.microsoft.com/solutions/msm.

Setup: Best Practices

Use the best practices in this section when you are performing the tasks to prepare for software update management.

Create production collections based on stable criteria

In general, using stable criteria to create collections for software update inventory and distribution will help to simplify all stages of the software update management process. Stable criteria you might use can include the installed client operating system version and service pack level, system role, or target organization.

Basing production collections on the operating system and service pack level, for example, ensures collection stability and minimizes excess generation of advertisement status messages. Use the same collections for distributing the scan component and distributing software updates, and create software update packages using the same criteria.

Create pre-production collections that include reference computers

The pre-production collection should include representative configurations of the operating system versions, line of business software, and other software running in your enterprise. You can create the pre-production collection automatically when you install the software update inventory tools by specifying a single computer to be placed in this collection; but afterwards, do not forget to modify the collection rules to include your other reference computers.

Provide a site-specific name for the scan component package

When you run the installer program for one of the software update inventory tools on the site server, you are prompted to provide a name for the package you are creating. This name should not be changed after the package is created. For this reason, it is important to choose a name that accurately distinguishes the tool and the site it manages when you view the package node for it in the SMS Administrator console.

Place computers running FAT file systems in their own collections

The **/cache** option for the scan component program can be used only on computers running the NTFS file system. You should place all computers that do not meet this criterion in their own collections, and advertise a custom scan tool program without the **/cache** option, to ensure that scan files are not tampered with before SMS runs them.

As a best practice, however, you should upgrade these computers to an NTFS file system if at all possible.

Ensure firewall/proxy access to the synchronization component

If you have a firewall that requires authentication, grant Guest access credentials to the IP address of the synchronization host, or specify a low-credentials domain user with Internet access and add information about this user account to the registry on the synchronization host. For more information, see the "Configure the Synchronization Host" section earlier in this chapter.

Co-locate the synchronization component and the scan component package source folder

When you are running the synchronization component in unattended mode, ensure that the computer hosting the package source folder for the scan component is also the computer that runs the synchronization component. This ensures that the synchronization component has proper credentials to access the package source folder. Be careful, however, to control the access to this folder to prevent unauthorized changes. For more information, see the "Task 1: Prepare the Package Source Folder" section earlier in this chapter.

Inventory Synchronization: Best Practices

Use the best practices in this section to ensure that the synchronization component of the software update inventory tools performs optimally.

Tune the synchronization component advertisement schedule

The synchronization component advertisement should run once a week for the Security Update Inventory Tool, and the day of its occurrence should be timed to the release of the security catalog update on the Microsoft Downloads Center. The Microsoft Office Inventory Tool for Updates can be synchronized less frequently — for example, once a month. For more information, see the "Scheduling: Best Practices" section later in this chapter.

Update distribution points on a schedule

When you configure the synchronization component for unattended use, the local computer account typically does not have credentials to update distribution points. In this case, you should turn off automatic distribution point refreshing for the synchronization component. Make sure that you schedule an update of the distribution points by using the procedure below. Refresh the distribution points daily if you are using reference computers. For more information, see the "Scheduling: Best Practices" section later in this chapter.

Periodically monitor the advertisement status for the synchronization component

Check the advertisement status summarizer for the synchronization component on a regular basis. Look for error or warning status messages that indicate download or runtime errors, access denied errors, or error number 12007 from authenticated proxy servers.

Software Update Inventory: Best Practices

Use the best practices in this section to ensure that the scan component of the software update inventory tools performs optimally and reliably.

Tune the scan component advertisement schedule

- Schedule the scan advertisement to the production collections every weekend for the Security Update Inventory Tool, every month for the Microsoft Office Inventory Tool for Updates.
- Schedule the scan advertisement to the pre-production (reference) collection daily, optimized to follow the update to distribution points.
- Do not link the scan advertisement schedule to the hardware inventory schedule. Configure the hardware inventory to use a simple schedule — once a week or every two weeks based on your existing policy and system loading. Use a more aggressive schedule for your collection of reference computers to monitor new and emerging issues in a timely manner.

Advertise the non-expedited program to the production environment

Do not use the expedited scan program in the production environment. A large-scale, expedited inventory results in a large amount of resynchronization transactions that are unacceptable in most production environments and should be avoided.

Advertise the expedited program to the pre-production collection

Using the expedited program in the pre-production collection helps you to respond quickly to emerging issues.

Do not use program dependencies in scan tool advertisements

The scan component of the software update inventory tools is set to run at regular intervals. If you specify a program dependency in this advertisement, the advertisement will run once and then subsequent occurrences of the advertisement will be skipped, because the dependent program was successful.

Disable the site-wide/per-program notifications for scan tool programs

The scan component runs as an unattended script on SMS client computers, and should remain as a background process that runs outside of the awareness of users.

Software Update Distribution: Best Practices

Use the best practices in this section to optimize the software update distribution process in your enterprise.

Create software update packages at the parent-site level of the hierarchy

In general, you should create and maintain your software update packages at the highest level in the SMS hierarchy from which you want to manage software updates. By creating and maintaining the packages at the highest level you ensure that there is uniformity in software update detection and authorization time throughout the site. You can then control package deployment at a more granular level by creating advertisements for the packages at child sites.

Reuse existing packages and collections when authorizing new software updates for distribution to stationary computers

A single software update package can contain multiple software updates, and these updates can be for multiple operating systems, versions, and client locales. At installation time, the Software Updates Installation Agent determines which software updates are applicable to a given SMS client computer, and installs only those updates. For this reason, it is best to organize your software update packages according to predetermined criteria, and then modify those packages when new software updates are authorized. When adding new software updates to a package, you can create a separate program for the new items to distribute them to the pre-production collection, and then merge the software updates into the main program after they have been tested.

Use a new package when authorizing selected software updates for distribution to mobile or remote computers

To conserve bandwidth for mobile computers and help increase compliance for critical software updates, consider creating separate packages for mobile computers that contain only the software updates that are authorized in the current week. Set the package advertisement properties on this Weekly New Updates package to download and run.

Organize software update packages and collections by operating system and service pack level

Create one software update package that contains all software updates for a specific operating system and service pack, and then create a collection that contains SMS client computers that are running that operating system and service pack. Do this for each operating system version and service pack level in your environment. When these operating systems reach the end of their supported lifetime, the software updates associated with them can easily be archived. This can also reduce the overall size of the packages making it easier for computers to download them prior to running them.

Migrate client computers to the Advanced Client.

The Advanced Client has several advantages over the Legacy Client for software update management. The Advanced Client can function more autonomously, and can issue reminders and provide enforcement capability that is independent of the advertisement schedule. For example, an Advanced Client can run an advertisement at the exact time software updates become required, even if the advertisement would not usually run for several more days. This allows a less frequent assignment schedule, provides greater end-user control over software update installation and system restarts, and can also reduce the overall processing that the site and clients undergo.

Group clients based on their SMS client version (Legacy Client or Advanced Client.)

Because the SMS Legacy Client does not support the persistent notification feature with its regular three-hour notifications, software update packages that you advertise to Legacy Clients require a more aggressive advertisement schedule (for example, daily as opposed to weekly). For this reason, it is best to place computers that are running the Legacy Client in their own collections wherever possible. This is a performance optimization to ensure that the Advanced Client computers receive a more appropriate advertisement frequency because they function more autonomously.

Advertise at least weekly to broad-based collections

You should set software update package advertisements to recur at least weekly. Only applicable updates will actually be installed, and then only after the software update installation settings you configure are honored.

Advertise daily in reference template mode to the pre-production collection

Although you must authorize at least one software update to accomplish this, gathering reference templates from the pre-production collection will facilitate the baselining strategy discussed earlier in this section. To do this, build one package of software updates for each baseline, and create a daily advertisement for these packages. This allows you to authorize software updates faster than the latency involved in using the normal inventory processing would otherwise permit. For more information, see the authorization list import feature.

Lock down the software update package source folder

When you authorize and distribute software updates with the Distribute Software Updates Wizard, you designate a package source folder in which to store the software update files that you have authorized. Because this folder contains the approved, verified, and tested versions of software updates for the software versions in use in your enterprise, it is part of your Definitive Software Library and should be protected. Steps to protect this folder include restricting access and performing regular backups. You should also make sure that you allocate adequate disk space for this folder.

For more information, see the “Task 1: Prepare the Package Source Folder” section earlier in this chapter.

Perform regular backups of the software update package source folder

The package source folder containing the software updates you have authorized, tested, and deployed in your organization contains value that increases with time as you add new updates to the package. For more information about backing up and restoring this folder, see Chapter 15, "Backup and Recovery."

Software Update Installation: Best Practices

Use the best practices in this section to control the way the Software Updates Installation Agent installs updates on SMS client computers. You configure these settings by using the three **Configure Agent Settings** pages in the Distribute Software Updates Wizard.

Use command-line options for each software update in a package

To avoid repeated system restarts and unnecessary user interruption, you should specify command-line options to suppress automatic system restarts and user interface for each software update in a package. At runtime, the Software Updates Installation Agent determines whether a system restart is needed by any of the software updates being installed, and manages any required restarts according to the settings you specified for the program/package.

Specify a user countdown of at least 30 minutes

You configure the countdown period in the **Wait <N> Minutes for User** setting on the second **Configure Installation Agent Settings** page of the Distribute Software Updates Wizard. The countdown period gives users time to save documents and review the list of software updates that are being installed. This is especially important for computers that are running the Legacy Client when the default action that is specified after the countdown is **Install updates** or **Perform restart**.

Specify the default action as Postpone for less urgent updates, Install for urgent updates

You configure the default action with the **After waiting** setting on the second **Configure Installation Agent Settings** page of the Distribute Software Updates Wizard.

Calculate the grace period from Time detected for mobile users, Time authorized for desktops

By specifying that the Software Updates Installation Agent calculate the allowable grace period from **Time detected**, rather than **Time authorized**, you can level the load on low bandwidth connections and prevent a situation where a software update might become required for all mobile clients at the same time. For desktop users, calculating the grace period from **Time Authorized** ensures faster response time. Also, when you are authorizing new updates, be sure to check the detection time listed for the software update in inventory if you are calculating the grace period from **Time Detected**. Be aware that a large lag between the time a software update is detected and the time that it is actually authorized might shorten or eliminate the grace period in this case

You can configure this setting in the settings that become available when you set the **Allow users to postpone installation for**: option on the third **Configure Installation Agent Settings** page of the Distribute Software Updates Wizard.

Use program dependencies in software update installation programs

When a new computer enters the environment, it is possible for the Software Updates Installation Agent to run on the SMS client computer before the scan component of the software update inventory tool has ever run. If this happens, the Software Updates Installation Agent will fail because there will be no cached version of the scan component for it to use for its just-in-time scanning. If you notice this situation happening based on the specific status message for this condition, consider changing the dependent program settings for the Software Updates Installation Agent program to ensure SMS runs the scan component first. Note that this does not force the scan component to run each time the advertisement runs; only the first time that the new client runs this advertisement.

End-User Experience: Best Practices

Use the best practices in this section to manage end-user experience and ensure fast uptake and low support costs.

Prepare end users with awareness and training prior to deployment

For best results and to avoid unnecessary calls to your support department, you should prepare end users to expect the software updates that you distribute to SMS client computers before you begin the distribution. This initial training can include appropriate screenshots and instructions.

Educate end users with branding and documentation attached to software update packages

The **Customize the organization** page of the Distribute Software Updates Wizard allows you to brand the software update package and include an optional .rtf file for display on SMS client computers during software update installation. You can use this file to help your end users understand the importance of the software updates being installed or to include instructions on scheduling the installation or required system restarts.

Note that if you are specifying a name for your organization in this page other than the default "Your system administrator," any text that you specify is not localized. Therefore you should ensure that this text is easily and intuitively recognized by all end users, regardless of locale.

Disable Automatic Updates for SMS client computers by using Group Policy

If automatic updates are enabled on a site where software updates are also being deployed with the SMS software update management components, users are likely to be confused, and it will also be difficult for you to perform service-level tracking of software update compliance. For this reason, it is best to disable the Automatic Update service.

Customize the software update description text for end users

By manually editing the software update authorization list (for example, PatchAuthorize.xml), you can provide richer and more detailed summary information for each software update than the pre-populated information that is provided by default. This information is displayed in the **Details** page when the software update installation notification appears on the client. To edit the authorization list, navigate to the package source folder and open the .xml file in a text editor such as Notepad. Edit the text between the <Summary> and </Summary> XML tags.

Customize software update advertisements to minimize user interaction

The **Environment** tab in the **Program Properties** page contains settings that allow you to specify that the program should run only when no user is logged on. Setting this property on your software update installation programs will increase the probability that users will not be interrupted by software update computer restarts. This is especially important for computers running the SMS Legacy Client.

Monitoring: Best Practices

Use the best practices in this section to monitor the various aspects of the software update management process.

Use SMS inventory data to query the vulnerability exposure for a software update

When responding to a new critical software update, you can use SMS hardware and software inventory to query clients according to criteria in the vulnerability matrix for that update. This is not necessary for deploying the software update, but it can be useful for determining the overall exposure to the vulnerability, and whether or how aggressively the software update should be deployed. For example, if the vulnerability only exists on computers that are running IIS, and no computers in a collection are running IIS, the software update deployment can be skipped for that collection.

Monitor status MIF text for run-time errors and summary data

In addition to monitoring the software update reports, you should develop a process for regularly monitoring the software update package advertisement status MIF files for errors and summary data. In the SMS 2003 release, status messages for summary and detail level status have been dramatically improved and are now complete status messages viewable with reports and the status message viewer in each SMS Server language.

Run compliance reports regularly

You should run regular reports to monitor the number of missing or installed updates, or updates with incomplete status, for each software update that is authorized. Similarly, reporting for software updates that are not yet authorized can facilitate easy deployment decisions. Try using the Dashboards feature of reporting to create a customized view of compliance, infrastructure health, and distribution status and include a link to this dashboard in your Internet Explorer Favorites.

Scheduling: Best Practices

The advertisements for the various software update management components are designed to run independently of each other. However, you can improve the performance, responsiveness, and reliability of your software update management process by optimizing the schedule of these advertisements.

For detailed information about the daily, weekly, monthly and as-needed tasks that are required to optimize software update deployment, see the white papers on software update management that are listed in Table 6.2.

Table 6.13 lists the tasks associated with software update management and their recommended frequencies.

Table 6.13 Software Update Management Tasks and Frequencies

Task	Performed by	Frequency
Security scan on SMS client computers	Automated, determined by package advertisement	Weekly
Office scan on SMS client computers	Automated, determined by package advertisement	Weekly
Synchronization (Security Update Inventory Tool)	Automated task on synchronization host	Weekly
Synchronization (Microsoft Office Inventory Tool for Updates)	Automated task on synchronization host	Weekly
Update Distribution Points (Security Update Inventory Tool)	Automated task, configured in package properties (see procedure below)	Weekly
Update Distribution Points (Microsoft Office Inventory Tool for Updates)	Automated task, configured in package properties (see the following procedure)	Weekly
Run Distribute Software Updates Wizard to modify Security update packages and add newly released or requested software updates	Administrator	Schedule determined by needs of IT organization. Should be performed at least weekly.
Run Distribute Software Updates Wizard to modify Office update packages and add newly released or requested software updates	Administrator	Schedule determined by needs of IT organization. Should be performed at least weekly.
Security updates distributed to SMS client computers (workstations)	Automated; determined by package advertisement	Daily/nightly depending on needs of enterprise.
Microsoft Office updates distributed to SMS client computers (workstations)	Automated; determined by package advertisement	Approximately twice a week, day or night, depending on needs of enterprise.
Security updates distributed to SMS client computers (servers)	Automated; determined by package advertisement	Schedule determined by server team. Should not configure automatic restarts.
Client hardware inventory regular schedule	Automated; determined by SMS hardware inventory configuration	Weekly for sites with more than 10,000 clients.

Table 6.14 shows a sample weekly schedule for these processes.

Table 6.14 Software Update Management Processes Sample Schedule

Task	M	T	W	Th	F	S	Su
Security Update Inventory Tool synchronization task						9:00 A.M.	
Update Distribution Points (Security Update Inventory Tool)						3:00 P.M.	
Security Scan on clients							Saturday night - Sunday morning
Microsoft Office Inventory Tool for Updates synchronization task							9:00 A.M.
Update Distribution Points (Microsoft Office Inventory Tool for Updates)							3:00 P.M.
Office Scan on clients							Sunday night - Monday morning
Run DSUW to modify Packages to add new security updates	Monday morning						
Office Update Advertisements (Workstations)		Daily (see below)		Daily (see below)			
Security Update Advertisements (workstations)	Nightly (see below)	Nightly (Run daily every two weeks)	Nightly (see below)	Nightly (see below)			

(continued)

Table 6.14 Software Update Management Processes Sample Schedule *(continued)*

Task	M	T	W	Th	F	S	Su
Security Update Installations (Servers)						Run on schedule determined by server team. No automated restart; restart schedule to be determined by server team.	
Client hardware inventory schedule					Weekly run date for SMS sites with more than 10,000 clients		

About Scheduling Software Update Installation Advertisements

The best schedule for running software update installation advertisements will vary depending on many factors. These include:

- The amount of user interaction you are allowing.
- The criticality of the updates contained in the package.
- Whether the client computers are running the Advanced Client or the Legacy Client.

Consider the following principles when setting the advertisement schedule:

- When advertising updates to computers that are running the Advanced Client, you can set the advertisement to recur less frequently (for example, once a day) and use the persistent notification feature.
- When advertising updates to computers that are running the Legacy Client, set the advertisement to recur more frequently to ensure that end users can see and respond to the notifications.

About Updating Distribution Points

A crucial step in staying current with your software update management process is the regular update of the software update inventory tools by the synchronization component. By default, this component works by automatically downloading the necessary files from the Internet, copying them to the package source folder for the scan component of the relevant tool, and then updating the distribution points with the updated package. However, when you configure this component to run in unattended mode, you must enable the update of the distribution points as a separate step.

► To modify package properties to update distribution points

1. In the SMS Administrator console, navigate to **Packages**:

```
Systems Management Serve
 └ Site Database (site code - site name)
  └ Packages
```

2. Right-click the package that you want to modify, and then click **Properties**.
3. In the **Package Properties** dialog box, click the **Data Source** tab, and then select the **This package contains source files** check box.
4. Under the **Source Directory** heading, perform the following tasks:
   - Click **Set**. In the **Set Source Directory** dialog box, specify the path for the package source files on the network.
   - Select the **Always obtain files from source directory** check box.
5. Select the **Update distribution points on a schedule** check box.
6. Click **Schedule** to specify how frequently to update the package data on distribution points.

   The default schedule for the update of distribution points is set to the current date and an interval of one day.

Click **OK** to save your changes and to close the dialog boxes.

Performance Considerations

This section describes performance considerations that you should be aware of when you use the software update inventory tools in your enterprise.

Processing Load Added to SMS Client Computers by the Software Update Management Components

CPU and disk utilization can increase when a software update is being installed on a client computer. The size and duration of the increase varies depending on the particular update. To obtain the exact size of the increase in processing load, it is recommended that you conduct pre-deployment testing for each update and determine the processing load increase by monitoring the test computers.

Inventory Data Considerations

The inventory data accrued for each software update can accumulate according to the number of software updates you are working with and the number of SMS client computers that are reporting the update.

Keep in mind the following information when you select updates and schedule inventory and installation cycles:

- Each software update creates approximately 2 KB of inventory data for each client that is reporting the update or reporting a change of state for the update.

  Note

  The above number is accurate at the time of this writing, but might vary in the future as software update inventory tools evolve. You can verify this number by inspecting a single software update instance inside the MIF files that are being generated by clients that are running the software update inventory scan tools.

- The initial software update inventory is large, since it creates a new data record for each software update that is applicable or installed on the client computer. Subsequent software update inventory scans will report only changes to the inventory data, such as newly available or applied software updates, and will generally be considerably smaller.
- History data for each software update also accrues, and will update the total SMS site database size on the server, when an update changes status from Applicable to Installed.

To help you calculate the effect that the software update inventory and distribution and installation of software updates will have on your system, multiply the numbers above by the number of clients you will be including in the inventory, and then plan the deployment of these tools accordingly.

One way to minimize the amount of inventory data passing through your system is to keep your client operating systems running the most current service pack version. For more information about this and other ways to optimize the performance of these tools, see the "Software Update Management Best Practices" section earlier in this chapter.

Scan Component Bandwidth Considerations

The scan component of the software update inventory tools consumes bandwidth at three different stages:

- The tools themselves consume bandwidth when they are initially distributed to client computers or are updated. The size of the bandwidth consumed in this operation depends on whether or not the client Msxml version needs to be upgraded. If not, you can calculate the size of the initial file copies by looking in the client cache folder at %*Windir*%\system32\vpcache\<*package ID*>.

  For clients that require an upgrade of their Msxml version before running the tools, the files for upgrading this application must also pass through the network during the initial installation of the scan component. You can calculate the size of this one-time event by adding up the .tmp file sizes for the Msxml application. For users running the Advanced Client and using Background Intelligent Transfer Service, these upgrade files can pass to the client in a background process.

- After the installation of the tool on the client, the local version of the software update catalog is updated (weekly by default). You can obtain an estimate of the size of this file by looking in the client cache folder for the software update inventory tool. For example, for the Security Update Inventory Tool, look at the 1033\mssecure.cab folder of the client cache folder.
- When the scan component runs, it sends software update inventory data. This is large for the initial software update inventory, and smaller for subsequent inventories. For a general estimate of the bandwidth consumed by this operation, see the "Inventory Data Considerations" section earlier in this chapter.

Scan Component Completeness Considerations

The accuracy of the software update inventory on SMS client computers is directly related to how current the local catalog of software updates is. To ensure that the scan component is using the latest software update information to create your inventory, do the following:

- Ensure that the software update catalog is current. If the synchronization component does not regularly download the updated version of the catalog, you risk the possibility of missing critical updates and creating an inaccurate inventory.
- If you have not configured the synchronization component to automatically update the distribution points, make sure you perform this step manually each time the synchronization component runs.
- Ensure that your process for using the synchronization component to download the latest database of software updates reflects the update schedule and frequency for that database. It is best to schedule the database download to occur as soon as possible after the database master copy is updated on the Web.

For example, the Security Updates Bulletin Catalog, MSSecure.xml, contains security update information that Microsoft updates regularly – once a week by default. Downloading this catalog on a weekly schedule (immediately following the Microsoft update) is generally optimal, and in most cases downloading the catalog more frequently does not provide any additional benefit or protection to your system. (Be aware, however, that Microsoft can update this file at any time if circumstances require it.)

For more information, see Table 6.14, "Software Update Management Processes Sample Schedule" earlier in this chapter.

Status Message Processing Considerations

An increase in status message processing is inevitable when you use the software update inventory tools to deploy software updates, because these tools generate status messages to track inventory and installation information. However, the size of the processing increase can be affected by your scheduling and configuration choices:

- The more frequently you schedule the inventory and installation cycles, the larger the increase in volume of status messages.
- If status message processing is a concern, then you can create status filter rules to eliminate the messages before they are replicated to the central site server.

Instantaneous Loading Considerations

Assignment schedules for updates usually activate at the same time, subject to Coordinated Universal Time (UTC, formerly Greenwich Mean Time) functionality. As a result, many clients can attempt to install software updates at the same time. This can cause system resource usage problems.

General Cumulative Effect of Scan Tools

The number of scan tools you use to create software update inventories has a direct relationship to the number of software updates, advertisements, and status messages using your system resources.

To minimize the problems associated with using multiple scan tools, you should manage the frequency with which you schedule inventory scans. As you use more scan tools, you should consider configuring the inventory scan cycles to match the download and synchronization cycle for the latest software update catalog.

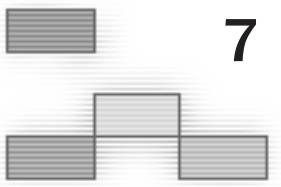
- To do this, determine when the new version of the catalog is published on the Web, and then schedule the catalog download to follow, as described in the "Scheduling: Best Practices" section earlier in this chapter.

Resolving Network Issues for Mobile Clients

Distributing software updates to mobile users can create network issues unless you plan for this scenario in advance. SMS 2003 offers many features that optimize software distribution to mobile users that are using the Advanced Client. See the "Software Update Management Best Practices" section earlier in this chapter for advice about managing mobile users, such as placing SMS Advanced Client computers in their own collections, which allows you to create custom advertisements for them to control whether the software updates in a package are required for mobile users and whether they are to be required if a local distribution point is not available.

CHAPTER 7

Creating Software Installation Packages with SMS Installer



Microsoft® Systems Management Server (SMS) 2003 includes SMS Installer, which is a tool that you can use to create software installation packages. These packages are known as SMS Installer-generated executable files, which are self-extracting files that contain everything that is necessary to install the software, including a script to control the installation. Although SMS Installer-generated executable files are created specifically for use on SMS clients, you can also post them to the Internet or package them on a CD or floppy disks. SMS Installer creates installation packages that can gather information about the current system, install and delete files, search for files, prompt users for information, and update both system files and the registry. You can customize the package to prompt the user for information or run unattended.

SMS Installer now includes the Windows Installer Step-up Utility (ISU). ISU is a command-line tool that migrates setup packages from the SMS Installer format to the Microsoft Windows Installer format. The resulting setup package is a Windows Installer setup package with an .msi file name extension. The new setup package can be run on any computer that supports Windows Installer. For more information about how to use SMS Installer, see the SMS Installer Help. SMS Installer also creates Windows Installer packages and can open SMS Installer-generated executable files.

This chapter begins with a description of how SMS Installer fits into the larger picture of software distribution. Then, the chapter describes how to create and modify installation scripts, test SMS Installer-generated executable files, and use these files to distribute software.

In This Chapter

- SMS Installer Overview
- Customizing Scripts with the Script Editor
- Testing SMS Installer-generated Executable Files

SMS Installer Overview

You run SMS Installer on a reference computer that is configured to match the target computers. Target computers are the computers that receive the installation package. For more information, see the "Reference Computer Preparation" section later in this chapter.

When you run SMS Installer, it gathers the necessary configuration data and automatically generates an installation script for the application. You can then use SMS Installer to modify the installation script. For example, you can modify the installation script to prompt users for specific information, translate user messages into different languages, or include support for restoring to a previous installation. You can also modify the installation script to run in the background without user input.

When the installation script is ready, you can use SMS Installer to convert the script into an SMS Installer-generated executable file or Windows Installer file that can be distributed to target computers and run. The Windows Installer packages can leverage the install on demand, repair, and advertise features that are provided by Windows Installer. You can distribute packages throughout your organization by using SMS advertisements, posting packages to the Internet or bulletin board system, or copying packages onto floppy disks or a CD. Setup files that are created by SMS Installer will run on Microsoft Windows® 98, Microsoft Windows NT® 4.0 (with the latest service pack), Microsoft Windows 2000, and Microsoft Windows XP.

SMS Installer Process

Because SMS Installer creates installation scripts, SMS Installer-generated executable files produce scripted installations. SMS Installer uses installation scripts to control the installation process. These installation scripts contain script commands that each perform a single action. These actions can be based on sophisticated conditions that are robust and flexible. Scripted installations make installing software both easy and less prone to error. Installation scripts can move files to the correct directories, prompt the user for information, give the user messages, and set registry keys and other values. You can specify which actions are performed by SMS Installer installation scripts. SMS Installer scripts can perform the following installation steps:

- Gather information from users
- Gather information about the current system
- Search for files
- Install and delete files
- Update .ini files and the registry

SMS Installer contains two user interfaces: Installation Expert and Script Editor.

Installation Expert

Use Installation Expert to automatically create a basic installation script on a reference computer, then use Script Editor to customize the script and add user prompts and other attributes.

Script Editor

Use Script Editor to view and edit an installation script generated by the Installation Expert, and then add user prompts or other attributes to your script. You can also use the script editor to create new installation packages.

SMS Installer also includes the options that are shown in Table 7.1.

Table 7.1 SMS Installer Options

Option	Description
Repackage Installation Wizard	A tool that replaces existing setup files with a customized script that you create by running the existing setup program and by creating a script from the changes that were made to the system during setup
Watch Application Wizard	A tool that creates a customized installation file for an application by noting the files that are used when you run the application and by creating a script from them
Compile	A program to create the self-executing file
Test	A program that tests the installation executable file without actually installing any files
Run	A program that runs the installation program on the reference computer
Compile as Windows Installer Package	A program to create Windows Installer (.msi) packages
Run as Windows Installer Package	A program that runs the Windows Installer (.msi) package
Uninstall Windows Installer Package	A program that uninstalls the Windows installer (.msi) package, if it is installed

The first time you start SMS Installer, Installation Expert opens. To switch between Installation Expert and Script Editor, click **Script Editor** or **Installation Expert** on the **View** menu. The user interface displayed at the end of your session appears the next time you start SMS Installer.

► To create an SMS Installer-generated executable file

1. Set up a reference computer on which you want to run the wizards to create the script.
   - If you are using the Repackage Installation Wizard to replace an existing setup program, the reference computer must be configured with exactly the minimum configuration that you require for your target computers.
   - If you are using the Watch Application Wizard to create a new setup program, there are no particular configuration requirements for your reference computer.

2. Use one of the wizards to create an installation script, and then edit and complete the script in Script Editor.

   You can also create the script entirely within Script Editor. Compile the installation script and files to create the compressed executable file, and then test the script by installing the files on a test computer.

   If you prefer to keep the existing setup program but want to add a script that executes it, you can create a wrapper script by using Script Editor.

3. Distribute the SMS Installer-generated executable file by using the following methods:
   - Distribute it automatically by using software distribution
   - Copy it onto a series of floppy disks
   - Copy it onto a CD
   - Post it to the Internet or a bulletin board system

SMS Installer Tasks

The process for creating an SMS Installer-generated executable file includes the following steps:

1. Determine if you need to use the Watch Application Wizard or the Repackage Application Wizard.
   - If the application already has a setup file, use the Repackage Application Wizard.
   - If the application does not have a setup file, use the Watch Application Wizard.
2. On the primary site server, unbundle the SMS Installer files.

   The files are packaged in such a way that they do not run unless SMS is installed.
3. To set up SMS Installer, copy the SMS Installer installation file (SMSInstl.exe) to the reference computer and double-click the **SMSInstl** icon.
4. To select the installation options you want, start SMS Installer and edit the SMS Installer attributes.

   There are 65 available options (script items), and you need to check each one carefully to ensure that they are set up the way you want. For information about each option, see the SMS Installer Help.
5. To automatically generate an installation script for the application, run the Repackage Application Wizard or the Watch Application Wizard.
6. Use Script Editor to modify the installation script. Usually, you must make at least a few modifications. For example, you can modify the script to prompt the user for information, send messages to the user, search for files, install and delete files, and update .ini files and the registry. Also, the wizard-generated scripts often benefit from adjustments. Test the script and examine it to see if some small changes make it more user-friendly and improve its performance.
7. Using the SMS Installer compiler, compile the SMS Installer-generated executable file.

8. Test the compiled SMS-generated executable file. SMS Installer has two test modes:
   - Test mode runs the installation program but does not install anything.
   - Run mode runs the installation program and installs the files.
9. Distribute the file.

All operating systems support long file names and the full Microsoft Win32® registry. You can create a single file or multiple files for posting packages to the Internet or bulletin board system or for copying packages onto floppy disks or a CD.

Installing and Starting SMS Installer

SMS Installer is only available by download and is not included with the SMS 2003 product. To download SMS Installer, see the Microsoft SMS Web site at http://www.microsoft.com/smserver/downloads.

You must first run the downloaded self-extracting file on a SMS 2003 primary site server. When SMS Installer has verified that your computer is a SMS 2003 site server, it copies SMS Installer with ISU installation files to the computer in the directory chosen. The default directory location is C:\SMS Installer Setup.

Use Microsoft Windows Explorer to navigate to the SMS Installer Setup directory. Copy the SMSInstl.exe file to the reference computer. To set up SMS Installer on the reference computer, double-click the **SMSInstl** icon. Or, you can share the SMS Installer Setup directory, map a drive to this share from the reference computer, and run SMSInstl.exe.

After you set up SMS Installer, you can either access the tools from the **Start** menu or use Windows Explorer to navigate to the C:\Program Files\Microsoft SMS Installer directory. When you find the directory, double-click the **SMSins32** icon. The 32-bit version can create 16-bit or 32-bit SMS Installer-generated executable files.

Creating Scripts with the Installation Expert

The Installation Expert creates installation scripts that control the installation process. These installation scripts contain script commands, each of which performs a single action. You can specify the actions that are performed by SMS Installer installation scripts by setting options in the **Installation Attributes** list.

The Installation Expert user interface includes Repackage Installation Wizard and Watch Application Wizard options.

These tools create automatically generated installation scripts. The scripts simply contain commands that place files in directories and set registry keys.

Running an Installation Wizard

After you copy the SMS Installer files to your reference computer and set up SMS Installer, you must edit all SMS Installer attributes. Then, create the installation script by choosing one of the follow methods:

- Use the Watch Application Wizard if a setup program for your application does not exist.

- Use the Repackage Installation Wizard if a setup program for your application exists, but you want to replace it.
- Use Script Editor if you want to create the script without running either wizard.
- Keep the existing setup program, but wrap it with an installation script.

  This approach is transparent to the user but allows you to customize the existing setup script. As a result, you retain the error-checking and branching that are built into many existing setup scripts. You must manually replace all the error-checking and branching in the installation script if you use the Repackage Installation Wizard.

Customizing Installation Attributes

Installation Expert is a flexible tool that can provide many ways to modify an installation script. Before you run either the Repackage Installation Wizard or the Watch Application Wizard, check the following installation attributes and ensure that they are set in the way that your installation requires:

- Installation Interface
- Application Files
- Runtime Support
- User Configuration
- System Configuration
- Advanced Configuration

Each of these attributes provides a number of script optimization options. You can find brief descriptions of these options in Table 7.2. For more information, see the SMS Installer Help.

To access these options, click **Installation Expert** on the **View** menu, and then double-click the attribute to display its dialog box.

Installation Interface Attribute

Table 7.2 lists and describes the functions of the **Installation Interface** attribute options. This attribute customizes the installation interface of the installation script that you are creating.

Table 7.2 Installation Interface Attribute Options

Option	Tab	Description/note
Single File or Floppy-Based Installation	Media	Compiles the source directory and installation script into a single file or divides the file into parts. Places the file into a directory with the same name as the installation script.

(continued)

Table 7.2 Installation Interface Attribute Options *(continued)*

Option	Tab	Description/note
Settings	Media	When you choose **Floppy-Based Installation**, you can set the file size.
Software Title	Application	Enter the name to be used in wizard dialog box titles, in the **Welcome** dialog box, and as the primary icon name. Do not use the word *installation* in the title because SMS adds it automatically.
Default Directory	Application	Name the top-level directory for the installation. In Windows 98 and Windows NT 4.0 installations, SMS places this file under Program Files.
Dialogs	Application	Selects dialog boxes for installation. Provides nine standard dialog boxes. These can be edited by selecting the **Dialog Templates** option on the **Edit** menu. This launches the Custom Dialog Editor, which is a separate application to help you manage your dialog boxes. In the Custom Dialog Editor, you can also add additional dialog boxes from the **File** menu.
Graphics	Graphics	Adds graphics to the installation and changes the graphics properties.
Status MIF	SMS	Sets up an SMS Status MIF file.

Application Files Attribute

You can use the **Application Files** attribute to add, modify, and sort all the components and files that SMS installs with the SMS Installer-generated executable file. To select the components that you want to install, use the **Components** tab.

Components are installed in the order that they appear on this tab. You can use **Add**, **Delete,** **Move Up**, and **Move Down** to create a list of the components that you want installed and the order you want them installed.

Use the **Files** tab to add, modify, and sort the folders and files you use in your installation. The user interface of the **Application Files Attribute Properties** dialog box consists of a top pane where you locate the folders or files to include in your script and a lower pane where you select a location to install these folders or files on the target computer.

Runtime Support Attribute

You can use the **Runtime Support** attribute to add additional components for Microsoft Visual Basic® and Microsoft Visual FoxPro®. The options on the **Visual Basic** tab are most useful when you create your own application with Visual Basic. In the **Options** dialog box, select components and add them to your installation. By default, only the **Uninstall Support** option is selected. You can edit several of the installation components by clicking **Details**.

- Use the **Visual Basic** tab to include Visual Basic run-time components.

  You must specify the directory where your Visual Basic system is installed so that SMS Installer can retrieve the run-time files. You can also specify the operating system; SMS Installer includes the run-time files for the operating system that you specify.

  The **Runtime Support** dialog box groups some of the Visual Basic run-time components so that a single check box includes all the files. You can include other single Visual Basic OLE custom controls (.ocx files) or dynamic-link libraries (DLLs) by using the **Files** dialog box of the **Application Files** attribute.

- Use the **Visual FoxPro** tab to select Visual FoxPro run-time component installation options.

  The **Runtime Support** dialog box groups some of the Visual FoxPro components so that a single check box includes all the files. You must specify the directory where your Visual FoxPro system is installed so that SMS Installer can retrieve the run-time files, or you can specify remote server support.

User Configuration Attribute

Use the **User Configuration** attribute to create program groups and associate icons with installable programs, to associate file types with viewing applications, to edit .ini files, and to change the registry of the target computer. Table 7.3 lists and describes the functions of the **User Configuration** attribute options.

Table 7.3 User Configuration Attribute Options

Option	Tab	Description/note
Select default group name for program manager group	Icons	Provide the name used as a submenu item.
Set up Associations	Associations	Set up associations between files with extensions unknown to the system and the applications used to view or run the files.
Modify .ini Files	INI Files	Modify .ini file settings.
Change registry on target computer	Registry	Set up changes that will be made to the registry of target computers during the installation.

System Configuration Attribute

Use the **System Configuration** attribute to add or change devices for operating systems other than Windows NT, to add or delete services in the installation script, or to cause the installation script to modify the Autoexec.bat or Config.sys file. Table 7.4 lists and describes the functions of the **System Configuration** attribute options.

Table 7.4 System Configuration Attribute Options

Options	Tab	Description/note
Modify the [386enh] section of the System.ini file	Devices	Add or delete devices or modify device properties.
Add services or edit their properties	Services	Add services to Control Panel or modify the service properties.
Modify Autoexec.bat file	Autoexec.bat	Produce a script that modifies the Autoexec.bat files of the target computer. You can choose to search for a line in Autoexec.bat where you can insert the new line. Make sure that the Autoexec.bat files of the target computers all contain the fields you search for.

(continued)

Table 7.4 System Configuration Attribute Options *(continued)*

Options	Tab	Description/note
Modify Config.sys file	Config.sys	Produce a script that modifies the Config.sys file of the target computer. You can choose to search for a line in Autoexec.bat where you can insert the new line. Make sure that the Config.sys files of the target computers all contain the fields you search for.

Advanced Configuration Attribute

Use the **Advanced Configuration** attribute to set advanced options such as screen, font, languages, patching, and global variables. Table 7.5 lists and describes the functions of **Advanced Configuration** attribute options.

Table 7.5 Advanced Configuration Attribute Options

Option	Tab	Description/note
Maximum Compression	Global	Select to choose a higher compression ratio for SMS Installer-generated executable files.
Control Installation Speed	Global	Select to slow the installation process on fast computers to allow the graphics to display.
No Installation Log	Global	Select to prevent creation of an installation log file. Prevents use of Uninstaller. Use this option if you are only copying files to the Windows, System, or Temporary directory.
Use Internal 3-D Effects	Global	Select to embed Ctl3d.dll into the installation executable file during installation. Presents dialog boxes in 3-D format. This option adds about 11 KB to the file size.

(continued)

Table 7.5 Advanced Configuration Attribute Options *(continued)*

Option	Tab	Description/note
ZIP Compatible	Global	Select to make the SMS Installer-generated executable file compatible with programs that read standard ZIP file format.
Replace in-use files	Global	Select to collect a list of files that must be replaced but are currently in use. Replaces files after rebooting the computer. Adds about 15 KB to the file size.
Convert CD-ROM to Floppy	Global	Select to change an existing installation script from a CD installation to a floppy disk installation.
Beep in New Disk Prompt	Global	Select to create an audio alert when a new disk is needed. Used in floppy disk installations only.
Suppress Reboot Message During Silent Installation	Global	Select to suppress reboot messages during an unattended installation.
Network Installation	Global	Select to reduce network traffic. Files that already exist on the computer are skipped, rather than reinstalled.
Use Verbose Output During MSI Compilation	Global	Select to receive all SMS Installer to Windows Installer migration details, including the status of each file that is converted.
Include Advertisement Support in MSI Output	Global	Select to add support for the Windows Installer install-on-demand (advertisement) option.
Installation Password	Global	Select an installation password. SMS Installer will prompt for this password during installation.

(continued)

Table 7.5 Advanced Configuration Attribute Options *(continued)*

Option	Tab	Description/note
Install Log Path Name	Global	Type a full path to a file that is used as a log file. Path characters must be alphanumeric.
Destination Platforms	Global	Select 16-bit and 32-bit platforms on which the software can be installed.
Progress Dialog Placement	Screen	Select where the **Copy** dialog box appears during installation.
Progress Bar Based On	Screen	Select an option for the progress bar. Possible values are: Position in Installation .exe (equal to the percentage of time for the percent done). Position in script (equal the percentage of time in each command regardless of relative time in each command). Percentage of selected files (equal the percentage of time for each file regardless of size).
Custom Progress Bar DLL	Screen	Browse to choose a custom DLL to be used for the progress bar instead of the actual **Progress** dialog box.
Center All Dialogs Over Progress Dialog	Screen	Select to center all dialog boxes and message boxes above the message bar.
Background Gradient	Screen	Select the size of the background window.
Title Bar	Screen	Select to display the title bar at top of the screen.
Hide Program Manager	Screen	Select to suppress Program Manager when icons are added or deleted.

(continued)

Table 7.5 Advanced Configuration Attribute Options *(continued)*

Option	Tab	Description/note
No Background Gradient	Screen	Select to eliminate the background gradient. This option is most useful when you have a background graphic.
Top Color	Screen	Select a color for the top of the gradient.
Bottom Color	Screen	Select a color for the bottom of the gradient.
Screen Preview	Screen	Displays the background window that you have created with your options.
Bold or Light Fonts	Font	Select normal fonts always, bold fonts always, or bold fonts for all platforms except Windows 98, Windows NT 4.0, and Windows 2000.
Message Box Font	Font	Select a font for message box text.
Point Size	Font	Select the point size of message box text.
Message Charset	Font	Select the character set number of message box text. If you translate your installation into Japanese, you must either set this field to 128 and set the Message Box font to MS Gothic or set the field to 0.
Languages	Languages	Select which languages to include in the file.
Default Language	Languages	Select the default language.
Japanese font name	Languages	Select the default name for the Japanese font.
Japanese Point Size	Languages	Select the point size for the Japanese font.

(continued)

Table 7.5 Advanced Configuration Attribute Options *(continued)*

Option	Tab	Description/note
Copy Default	Languages	Specify the default font name and point size. If you select this box, messages appear in the default language when messages have no translation into the current language.
Always Prompt	Languages	Select to have SMS prompt the user for a language when the script is compiled and language messages are missing.
Prompt to Save	Options	Select to be prompted to save the file each time a new SMS Installer-generated executable file is created.
Run in Manual Mode	Options	Select to be prompted to select the locations for certain files each time that you run your installation.
Show Toolbar Tips	Options	Select to make ToolTips part of your installation.
Show Status Bar Tips	Options	Select to make status bar tips available.
Append New Items	Options	Select to append new items after the currently selected action, rather than before the action, as you edit your installation script.
Suppress Version Error	Options	Select to suppress version checking during the **Install File** action, when a file that does not have a version resource is detected.
Background Processing	Options	Select to enable your system to process background tasks during the compile process.

(continued)

Table 7.5 Advanced Configuration Attribute Options *(continued)*

Option	Tab	Description/note
Smart Create	Options	Select to detect if the date or time of an SMS Installer-generated executable file has changed and to create a new file only if the date or time has changed.
Fast Create	Options	Select to speed up the installation-creation process by copying the compressed version of files from a previous installation script to the new file. If the size or date of a file has changed, the file is replaced.
Exclude DLLs	Options	Specify DLLs to exclude from dependency checking in the Watch Application Wizard.
Installation .exe name	Settings	Type a full path for the executable file or browse to the directory.
Language INI Name	Settings	Type a path (or browse) to the .ini file that contains the language translations for the installation.
Setup Icon Path	Settings	Type a path (or browse) for the Setup file icon (16-bit only).
Dialogs Directory	Settings	Type a path (or browse) to the directory that contains the dialog boxes.
Temp Files Directory	Settings	Type a path (or browse) to the directory that contains the temporary files.
Do Not Create Patching Updates	Patching	Click to provide copies of entire files rather than creating patches.
Create Patching Updates	Patching	Click to provide patches rather than creating copies of entire files.
Error Checking	Patching	Select the level of error messages.

(continued)

Table 7.5 Advanced Configuration Attribute Options *(continued)*

Option	Tab	Description/note
Patch Threshold	Patching	Select a percentage of a file that is replaced where patching occurs below a particular limit but the entire file is replaced above this limit.
Maximum Memory	Patching	Select a size, in kilobytes, to limit the amount of memory that can be used to create a patch.
Maximum Patch Compression	Patching	Select to enable maximum compression for the patch file.
Add	Compiler Variables	Opens the **Compiler Variable Settings** dialog box so you can add another variable to the list.
Delete	Compiler Variables	Deletes the selected variable.
Properties	Compiler Variables	Click to display properties of the selected variable. Opens the **Compiler Variable Settings** dialog box.
Compiling from Command Line	Compiler Variables	Select to prompt the end user to provide compiler variables when compiling from the command line.
Compiling from IDE	Compiler Variables	Select to prompt the end user to provide compiler variable when compiling from an integrated development environment (IDE).
Do not create a Code-Signed Installation	Signing	Select to create an unsigned installation.
Create a Code-Signed Installation	Signing	Select to create a code-signed installation.
Web URL	Signing	Add a Web URL for this installation.
Descriptive Name	Signing	Provide a descriptive name for the Web URL.
Credentials File	Signing	Select a credentials file for the URL.

(continued)

Table 7.5 Advanced Configuration Attribute Options *(continued)*

Option	Tab	Description/note
Private Key	Signing	Select a private key for the credentials file.
CAB File	Signing	Choose whether to create a .cab file. If you create a .cab file, SMS Installer places the .exe file in the .cab file. Optionally, you can provide the contents of a Setup.inf file.
File Version	Version	Type the version number of the setup program.
Description	Version	Type a short description of the setup program. You can enter up to 256 characters.
Copyright	Version	Type the copyright information for the setup program. You can enter up to 256 characters.
Other Version Info	Version	Provides additional information about the setup program. This includes Company Name, Internal Name, Language, Legal Trademarks, Original File Name, Product Name, and Product Version. You can modify the information by highlighting the item in the **Item Name** box and then modifying the value in the **Value** box.

Repackage Installation Wizard

The Repackage Installation Wizard replaces an application's existing setup program with a new one that you create. The wizard produces the basic script. Using Script Editor, you can add any error checking, user interaction, branching, additional files, and registry key changes.

The **Repackage** button in the **Installation Expert** dialog box starts the Repackage Installation Wizard. The Repackage Installation Wizard performs the following tasks:

1. Scans the reference computer
2. Runs Setup for the application

3. Scans the computer again to detect all the changes that occurred during the setup process
4. Uses the detected changes to create the installation script

When you run the Repackage Installation Wizard, you specify the path of the application's setup program. You can also specify command-line options to use when Setup runs and modify which directories, files, and registry keys are scanned.

During the repackaging process, SMS Installer helps you to configure or otherwise modify the application by:

- Modifying the list of files and directories that are scanned.
- Modifying the list of registry key changes to include in the script.

You can also customize dozens of installation script options by modifying SMS Installer installation attributes. Before you run the Repackage Installation Wizard, see the "Customizing Installation Attributes" section earlier in this chapter and change any of the default attributes that your application requires.

Reference Computer Preparation

The first step in preparing an SMS Installer-generated executable file is to prepare the reference computer that you use to set up and run the application.

When you configure the hardware and software, it is recommended that the reference computer be identical to the target computers on which the installation executable file will run. Otherwise, the installation script that is created on the reference computer might not detect important files and might fail to install them on the target computer.

Caution

Although it is recommended that the reference computer be identical to the target computers in most respects, it must not be an SMS client or server. If it is an SMS client or server, configuration data might be transferred to the target computers and interfere with normal SMS operation.

Before running the Repackage Installation Wizard on the reference computer, it is recommended that you verify the following:

- The reference computer and all target computers have the same operating system installed. They should also have the same version number and service pack.
- The reference computer and all target computers have the same applications installed. In general, unless there is a specific dependency on an existing application by the repackaged application, make sure that the reference computer only has software that is needed directly by the repackaging process.
- The reference computer and all target computers have the same hardware installed. This point is especially important when the software makes configuration changes in target computer hardware settings.

Be sure to use a reference computer that satisfies the minimum configuration that you require to install your software. Many applications share files. If the repackaging process determines that these shared files were not added to the reference computer, they are not included in the SMS Installer-generated executable file. As a result, the repackaged application might not run correctly. For example, if you want to repackage Microsoft Word, and Microsoft Excel is installed on the reference computer, some of the shared DLL files and the files in the MSAPPS directory might not show up in the installation script. As a result, if Excel is not already installed on the target computers, the repackaged version of Word does not install completely and might fail to run correctly.

Note

Whenever you repackage additional files for other applications, you must rebuild the reference computer with clean copies of the necessary software. Otherwise, your reference computer may not reflect an adequate starting point and the repackaging process may not detect configuration changes.

Running Repackage Installation Wizard

The Repackage Installation Wizard automates the process of creating an SMS Installer-generated executable file.

To run the Repackage Installation Wizard

1. On the **Start** menu, point to **Programs**, point to **Microsoft SMS Installer**, and then click **Microsoft SMS Installer 32**.
2. If SMS Installer opens in Script Editor, click **Installation Expert** on the **View** menu.
3. Click **Repackage**.
4. In the **Repackage Installation** dialog box, type a complete path to the installation program in the **Installation Program** box.

   It is recommended that this full path not contain any command-line options or arguments. If you prefer to select a program on your computer, click **Browse**.
5. In the **Command-Line Options** box, type any command-line setup options that you want for your setup program.
6. In the **Directory** box, under **Sub-Tree**, indicate whether to scan subdirectories of the directories you have chosen.

   To modify how SMS Installer scans the reference computer, click **Change.** Use the **Files/Directories** and **Registry Keys** tabs to modify the settings in the **Repackage Advanced Settings** dialog box.
7. To start the Repackage Installation Wizard, click **Next**.

   The Repackage Installation Wizard completes the first scan of the reference computer and then starts the setup program that you specified. To complete the setup, follow Setup screen instructions and complete the setup as you want it to be completed on the target computers.

8. When the setup program is complete, you can make any additional changes that you want in your installation script to the application or reference computer. After you make any changes, click **Next** to complete the repackaging process.
9. To return to the Installation Expert, click **Finish**.
10. To name your installation script and save it in a directory, click **Save As** on the **File** menu, and then type a name.

Configuring Repackage Installation Wizard

When SMS Installer scans the reference computer during the repackaging process, SMS scans up to 32 levels in a directory tree and up to 64 levels in a registry tree. The files and script items that SMS Installer includes within a script are subject to the following limits:

- A script can include up to 5,888 files. SMS adds one Install File script item for each file.
- A script can include up to 8,192 script items (up to 5,888 Install File script items).

When you configure SMS Installer to repackage an application, consider the following issues:

Data conversion

If the original setup program upgrades or modifies data files, such as user database files, the Repackage Installation Wizard fails to capture the conversion. As a result, the SMS Installer-generated executable files are not installed correctly on the target computers. If the original setup program includes data conversion, do not use the Repackage Installation Wizard.

Hardware scans

If the original setup program detects hardware and the target computers do not have hardware and drive configurations that are identical to the reference computer, a repackaged SMS Installer installation might fail. If you cannot be sure that the reference computer and target computers have identical hardware and drive configurations, you can work around this constraint. Either modify the script after it is produced to query users for the necessary information or do not use Installation Expert. You might want to use Script Editor to prepare a script that runs the original setup file.

Shared network files

If the original setup program modifies shared or network files, test the repackaged installation program carefully and modify it by using Script Editor, if necessary. The Repackage Installation Wizard is very flexible, but if it tries to modify shared network files the installation might fail. If the Repackage Installation Wizard even references network files, the installation might fail. If you think this could be a problem for your installation, conduct extra testing to ensure that the repackaged installation file runs on all clients and under all user accounts.

Custom Configuration for Repackaging Scans

By default, during the repackaging process, the Repackage Application Wizard scans the drive where the Windows operating system is installed. This scan includes all directories, files, and registry settings that are changed by the installation. However, Installation Expert cannot detect which changes are directly related to the installation. While the installation program runs, the system might update certain temporary files (.log or .tmp) and certain registry keys that are unrelated to the application installation. It is recommended that you do not include these updates as part of your installation script.

You can configure SMS Installer to scan additional drives and also to ignore certain directories, files, and registry keys. For example, to prevent temporary file updates from appearing on your target computers when they actually occurred on the reference computer, you can specify that SMS Installer ignore certain log or temporary files. You can also remove from the script any registry keys that might be changed but are not part of the installation. For example, the installation might change a Dynamic Host Configuration Protocol (DHCP) release and renew with a new TCP/IP address or recently used documents in the **HKEY_CURRENT_USER** subtree. It is recommended that you do not include changes unrelated to the installation in the installation scripts.

You can decide which directories SMS Installer scans. Remember that the fewer directories that are scanned, the faster repackaging occurs. However, if you are not sure which directories the setup program writes to, scan them all.

To configure the Repackage Installation Wizard to add or remove files and directories from the scan list, click **Change** in the **Directory/Subtree** box in the Repackage Installation Wizard. Then, navigate to the **Files/Directories** tab in the **Repackage Advanced Settings** dialog box.

- To add a directory, select the directory that you want SMS Installer to scan in the **Directory** box, and then click **Add**.
- To delete a directory, select the directory that you do not want SMS Installer to scan in the **Directory** box, and then click **Delete**.
- To select a file that you want SMS Installer to ignore, click **Add** in the **File Name** box, and then complete the dialog box.
- To remove a file from the list of files that you want SMS Installer to ignore, select the file in the **File Name** box, and then click **Delete**.

To configure SMS Installer to ignore registry keys in the repackaging process, navigate to the **Registry Keys** tab in the **Repackage Advanced Settings** dialog box.

- To add a subtree to the list of subtrees that you want SMS Installer to ignore, locate and select the registry subtree, and then click **Add Tree**.
- To remove a subtree from the list of subtrees that you want SMS Installer to ignore, select the subtree, and then click **Delete**.

- To add a registry key that you want SMS Installer to ignore, locate and select the registry subtree that contains the key, select the key in the box where it appears, and then click **Add Value**.
- To remove a registry key from the list of registry keys that you want SMS Installer to ignore, select the value, and then click **Delete**.

Watch Application Wizard

The Watch Application Wizard is most useful when you want to create an SMS Installer-generated executable file for an application that has no existing setup program. This wizard runs an existing application and notes the files that are used. The wizard adds these files to an installation script for the application. You can modify the installation script and compile it into an SMS Installer-generated executable file that you can deploy throughout your organization.

You can also use the Watch Application Wizard to verify that the Repackage Installation Wizard has captured all the files that are necessary for an application. For example, suppose that a developer that is using Visual Basic creates an application. The developer includes all the new files in the setup process but is not aware of support files that are called automatically by Visual Basic and its run-time components and that are necessary to the setup program. In the Repackage Installation Wizard, the repackaging process is completed successfully on a computer that has Visual Basic, but the installation files list is incomplete for a target computer without Visual Basic. The Watch Application Wizard allows you to discover these additional files so you can add them to the installation script manually.

The Watch Application Wizard does the following tasks in order:

1. Runs an existing application on the reference computer, noting the files used by the application
2. Uses the list of files to create an installation script for the application

Running the Watch Application Wizard

You run the Watch Application Wizard on a reference computer on which the existing application is already installed. This computer can have any configuration. The Watch Application Wizard runs the application and notes the DLLs, OLE custom controls (.ocx), and Visual Basic Custom Controls (VBXs) that are used. When complete, these files are added to an installation script for the application. You can then modify the script and compile it into an SMS Installer-generated executable file.

As you start the Watch Application Wizard, be sure to specify the Visual Basic configuration options that you want on the **Visual Basic** tab in the **Runtime Support** dialog box. For information, see the "Runtime Support Attribute" section earlier in this chapter. If there are DLL files that you want excluded from the **Watch** function report, you must use the **Options** tab in the **Advanced Configuration** dialog box to exclude them. For more information, see the "Advanced Configuration Attribute" section earlier in this chapter.

► To run the Watch Application Wizard

1. On the **Start** menu, point to **Programs**, point to **Microsoft SMS Installer**, and then click **Microsoft SMS Installer 32**.

   If SMS Installer opens in Script Editor, click **Installation Expert** on the **View** menu.
2. In the **Installation Expert** dialog box, click **Watch**.
3. In the **Watch Application** dialog box, specify the path to the application.
4. Run the application and use all of the program features of the application.
5. When you have run all the possible commands for the application, click **Finish**.

   The files that were accessed are listed in the installation script in Script Editor. They are also listed in the **Application Files** installation attribute on the **Files** tab in the **Installation Expert** dialog box.

Customizing Scripts with the Script Editor

After you create the basic installation script with Installation Expert, you can edit the script by using Script Editor. Script Editor is a flexible, powerful tool that you can use to create variables and branching within the installation script. In addition, you can use Installation Expert to add the following customized functions:

- Prompt users for information
- Add files and directories to a script
- Include other scripts
- Provide uninstall and rollback support
- Change SMS Installer messages
- Change the registry
- Register third-party applications and controls
- Add your application to **Add or Remove Programs** in Control Panel
- Run programs at startup
- Provide conditional flow control of script execution

Many customized functions can be inserted by using the Script Editor actions, or you can add them to the script by configuring **Installer Attributes** in Installation Expert. For example, you can use either method to provide uninstall and rollback support, and to add your program to **Add or Remove Programs** in Control Panel. If you modify a graphical user interface, Installation Expert adds the script items to your installation script. You can also add or change them manually by using Script Editor.

If you plan to use Installation Expert at any point during the script building process, it is recommended that you use Installation Expert to create the basic installation script. By using this approach, you can switch between Installation Expert and Script Editor without losing customization due to the conversion. If you create the script with Script Editor and then switch to Installation Expert, some script items might be lost.

- To edit a line of a script, double-click it. If the item can be edited, a dialog box with the properties of the item appears.
- To add a line to a script, select the line following the position where you want to add the item. Then, double-click the item that you want to add in the **Actions** list or drag the item to the place in the script where you want it.

Script Editor User Interface

Script Editor includes an **Actions** list and an **Installation Script** box containing your installation script.

Script Editor Options

Script Editor contains the following options that you can use when you create or modify installation scripts:

Title

Use this box to enter the text that is displayed in the title bar while the installation runs.

Event

Use this box to select the script for the current setup file. Choices include:

- **Mainline**. The script that runs during the installation.
- **Exit**. The script that runs when the installation is successfully completed or when the Mainline script contains an **Exit Installation** script item. For example, this script can prompt users to run the program that was just installed.
- **Cancel**. The script that runs when the installation is not completed successfully or when the user clicks a **Cancel** button in a setup dialog box. For example, this script can perform cleanup tasks.

Language

The language of the current setup script. You can add more languages if you are creating a multilanguage script. However, you can add only the languages that you selected when you installed SMS Installer. If you want to add more languages, you must reinstall SMS Installer and choose the additional languages you need.

Actions

A list that contains all the possible actions that the installation script can perform. To display the dialog box that is associated with a script item, double-click the action that you want. To insert the action in the script above the selected line, click **OK**.

Installation Script

The current installation script. To display the dialog box associated with a script item, double-click the action that you want.

Script Editor Menus

Script Editor contains four menus:

File

Includes a function to copy the SMS Installer-generated executable file to floppy disks.

Edit

Includes functions to edit the locations of source directories, dialog box templates, and SMS Installer messages within the installation script.

View

Includes a toggle between SMS Installer, Installation Expert, and Script Editor.

Build

Compiles, tests, runs, and debugs the installation script. It also includes options to migrate compiled SMS Installer Setup packages to the Windows Installer format. You can compile as a Windows Installer package, run as a Windows Installer package, or uninstall a Windows Installer package. For more information about how to migrate compiled SMS Installer Setup packages to Windows Installer format, see the SMS Installer Help.

Installation Script Variables

Script variables hold information about the installation that is being performed. You use these variables to retain the information that is gathered from users about where to place files. They are also used to hold information about which files that users want to install. In addition, a number of predefined variables contain information about the target computer on which you are installing software.

In script commands, variables have two roles: destination variables and variable references.

Destination variable

When a script command places information into a variable, the variable is a destination variable. You must specify the name of the variable to use. The variable name must:

- Begin with a letter.
- Include only numbers, letters, and the underscore (_) character.
- Contain 14 or fewer characters.

Variable reference

When you want to use the value that is in a variable, place the variable name within percent signs (%). This is called a variable reference.

For example, if you want to set the value of the variable DEFAULTDIR to C:\Temp, use the **Set Variable** script command. Make sure that the **Variable** field contains DEFAULTDIR and set the **New Value** field to C:\Temp.

To set the value of DEFAULTDIR to be the same as the WIN variable (which contains the Windows directory name), set the **Variable** field to DEFAULTDIR and the **New Value** field to %WIN%. The percent signs indicate that you are using the value of the WIN variable.

Note

Because the percent sign is used to signify the value of a variable, if you want a percent sign in the message text of a script command, you must use two percent signs together. For example, to display a message to users that they have completed half of the installation, use the following text: "The installation is 50% %complete."

Predefined Variables

SMS Installer creates and defines variables at the beginning of installation. You can use the variables in your installation scripts.

Table 7.6 lists and describes the function of the predefined variables.

Table 7.6 Predefined Variables

Variable	Description
WIN	Contains the path of the Windows directory (usually C:\Windows).
SYS	Contains the path name of the Windows System directory (usually C:\Windows\System).
SYS32	Contains the system directory for Win32 files under Windows NT (usually C:\Winnt\System32).
TEMP	Contains the directory that temporary files can be placed in. This variable is useful for placing DLLs before you call their functions.
INST	Contains the directory from which the SMS Installer-generated executable file is run. This variable can be useful if you want to display a Readme.txt file that is located on the same disk as the SMS Installer-generated executable file.
CMDLINE	Contains the command-line options that were passed to the SMS Installer-generated executable file.
LANG	Contains the language that users selected in a multilanguage installation.

(continued)

Table 7.6 Predefined Variables *(continued)*

Variable	Description
FONTS	Contains the directory on the target computer in which fonts are installed.
PASSWORD	Holds the installation password for a password-protected installation package.
PROCEXITCODE	Contains the exit code of the last process called by using the **Execute Program** script item with the **Wait for Program to Exit** option selected.

Creating Variables

During the installation, you can create variables that SMS uses to perform certain functions. Use the **Set Variable** action in the Script Editor **Actions** list to create such variables or use the **prompt** command. For example, you can create the following useful variables.

HELPFILE

Specifies the Help file that is displayed during installation when the user clicks **Help**.

RESTART

Restarts Windows at the end of an installation. It is set automatically.

DOBACKUP

Creates a backup of all files that changed during an installation.

BACKUPDIR

Specifies the directory in which to place backed-up files.

Table 7.7 lists and describes the functions of the options in the Script Editor **Items** list.

Table 7.7 SMS Installer Script Editor Items

Option	Description	MSI compatible
Add Device to System.ini	Adds or modifies entries in the [386Enh] section.	Yes
Add Directory to Path	Appends the specified directory to the PATH environment variable.	Yes
Add ProgMan Icons	Manages icons and groups in Program Manager and on the **Start** menu.	Yes
Add Text to Installation Log	Adds remarks to the installation log file.	No

(continued)

Table 7.7 SMS Installer Script Editor Items *(continued)*

Option	Description	MSI compatible
Add to Autoexec.bat	Adds or replaces commands and environment variables in the Autoexec.bat file, except for the PATH environment variable.	Yes
Add to Config.sys	Adds device drivers to Config.sys.	Yes
Allow Floppy Disk Change	Changes the floppy disk so that you can run another executable file during the installation process.	No
Browse for Directory	Provides a generic directory browse dialog box.	Yes
Call DLL Function	Calls Win16 and Win32 DLLs.	Yes
Check Configuration	Checks a finite set of configurable items on the target computer, such as the operating system and amount of memory.	Yes
Check Disk Space	Verifies that enough disk space is available on the target computer to complete the installation.	Yes
Check If File/Dir Exists	Verifies that a file or directory exists on the target computer.	Yes
Compiler Variable	Provides if/then/else logic for compiler variables.	Yes
Configure ODBC Data Source	Creates and configures an Open Database Connectivity (ODBC) data source.	Yes
Copy Local Files	Copies uncompressed files from your installation disk to the target computer.	Yes
Create Directory	Creates an empty directory on the target computer.	Yes
Create Service	Creates a service on a target computer that is running Windows NT.	Yes

(continued)

Table 7.7 SMS Installer Script Editor Items *(continued)*

Option	Description	MSI compatible
Create Shortcut	Creates a shortcut on the Desktop or **Start** menu for target computers that are running Windows NT.	Yes
Custom Dialog Box	Use to create custom dialog boxes to display and request information during the installation.	Yes
Custom Graphics	Use to create and edit graphics that are displayed during the installation.	Yes
Delete File(s)	Deletes files and directories on the target computer.	Yes
Display Graphic	Displays bitmap files in the background during the installation.	Yes
Display Message	Displays a message to the user and captures the user's response.	Yes
Display Readme File	Creates a dialog box that is used to display the contents of any text file.	Yes
Edit .ini File	Creates or edits an .ini file on the target computer.	Yes
Edit Registry	Edits the system registry.	Yes
Else Statement	Provides the FALSE condition to your script's logic.	Yes
End Block	Ends a logical block of script items that begin with a start block (if/else) or a WHILE loop.	Yes
Execute Program	Helps you execute another program (outside of the installation) during the installation process.	Partial. DDE functionality in SMS Installer is not supported through Windows Installer.

(continued)

Table 7.7 SMS Installer Script Editor Items *(continued)*

Option	Description	MSI compatible
Exit Installation	Terminates and exits the installation.	Partial. MIF generation is handled internally in Windows Installer, so no customization is possible.
Find File in Path	Finds the first occurrence of a file in a directory tree or in the PATH environment variable on the target computer.	Yes
Get Environment Variable	Loads the value of the environment variable into a script variable.	Yes
Get Name/Serial Number	Creates a dialog box to request up to three pieces of information from the user.	Yes
Get ProgMan Group	Creates a dialog box that displays a list of Program Manager groups on the target computer and helps the user to select from the list or enter a new group.	Yes
Get Registry Key Value	Retrieves data values from the system registry.	Yes
Get System Information	Retrieves system information from the target computer, such as Windows version number and file size.	Yes
Get Temporary Filename	Creates a unique temporary file name in the \temp directory on the target computer. You must create the file yourself by using the variable to which the file name is assigned.	Yes
If/While Statement	Controls the flow of logic in your script.	Partial. The Windows Installer service does not reproduce timing or delay loops. Using complex If/While statements force the use of MSI nesting, which does not allow Windows Installer's advertisement

(continued)

Table 7.7 SMS Installer Script Editor Items *(continued)*

Option	Description	MSI compatible
Include Script	Incorporates other scripts into your script at compile time.	Yes
Insert Line into Text File	Adds lines of text to new or existing text files.	Yes
Install DirectX	Installs DirectX® drivers on the target computer.	No
Install File(s)	Compresses files that are installed on the target computer into the installation executable file.	Yes
Install MMC Snap-in	Compresses the Microsoft Management Console snap-in DLL into the SMS Installer-generated executable file.	Yes
Install ODBC Driver	Adds a driver name and driver attributes to the Odbcinst.ini file and to the system registry.	Yes
Modify Component Size	Modifies the amount of space that SMS Installer calculates for a given component.	No
Open/Close INSTALL.LOG	Opens, closes, or resumes writing to the log file.	No
Parse String	Searches a string for a pattern and splits the string into two new strings based on the position of the pattern.	Yes
Play a Multimedia File	Plays audio and video files during the installation.	Yes
Prompt for Text	Creates a dialog box to prompt the user for a single line of text.	Yes
Radio Button Dialog Box	Creates a dialog box that prompts the user to select from a set of options.	Yes
Read INI Value	Reads an item entry from an existing .ini file on the target computer.	Yes

(continued)

Table 7.7 SMS Installer Script Editor Items *(continued)*

Option	Description	MSI compatible
Read/Update Text File	Reads and updates lines of text in a text file on the target computer.	Yes
Read/Write Binary File	Reads from a file and writes to a file in binary mode.	Yes
Register Font	Registers fonts that you have copied to the target computer.	Yes
Remark	Adds comments and white space to your script.	No
Remove From System.ini	Removes (comments) entries in the [386Enh] section.	Yes
Rename File/Directory	Renames a file or directory on the target computer.	Yes
Search for File	Locates a file on the target computer.	Yes
Select Components	Creates a component selection dialog box.	Yes
Self-Register OCXs/DLLs	Registers .ocx and DLL files.	Yes
Set File Attributes	Sets the file attributes of a file or group of files.	Yes
Set Files/Buffers	Modifies the FILE and BUFFER settings in the file.	Yes
Set Variable	Creates a script variable and modifies the content of a script variable.	Yes
Sleep	Pauses the installation process for a specified amount of time.	Yes
Start/Stop Service	Starts or stops a service.	Yes
Win32 System Directory	Gets the path to the target computer's system directory.	Yes
Wizard Block	Controls the logical flow of wizard dialog boxes in your script.	Yes

Using an Installation Script to Wrap an Existing Setup

Instead of repackaging an installation, you might want to keep the original installation file but add some user interaction or run the script with certain command-line options. To do this, open SMS Installer in Script Editor. Insert the **Execute Program** script item and run the setup program, using command-line options to customize the install.

If you do not know which command-line options are available, many programs include a short Help file that describes the options. You can try typing the program command at the command prompt followed by a question mark. Often, this will list the available options. If not, you can contact the manufacturer of the program to see if it can be run with command-line options.

After you have typed the command-line option, surround it with any other script items that you need. Then, compile the installation file and test it. The original setup program is not repackaged with the SMS Installer-generated executable file. You must distribute the original application files in the same directory with the SMS Installer-generated executable file.

Unattended Setup Script

You can use SMS Installer to create a file that runs unattended on target computers. You can do this task by running an existing setup script and by using command-line options. Or, you can repackage the original setup file so that it runs unattended.

To run the existing script and use command-line options, see the "Using an Installation Script to Wrap an Existing Setup" section earlier in this chapter. To run Setup unattended, use the **/s** switch. This switch suppresses all the dialog boxes that are part of the normal SMS Installer script.

If you are repackaging an application with a setup program that would usually require the user to restart the computer during the setup procedure, it is recommended that you suppress the restart message. In Installation Expert, click **Advanced Configuration**, and then select **Suppress Reboot Message During Silent Installations** on the **Global** tab.

Testing SMS Installer-generated Executable Files

After you compile the SMS Installer-generated executable file, it is recommended that you test it. Testing can show you what the installation will look like when it is run on a target computer. Because there are so many opportunities for customization with SMS Installer, it is particularly important to test the package thoroughly and make sure no changes, such as suppressing a dialog box, are necessary.

SMS Installer provides two modes for testing SMS Installer-generated executable files:

- The test mode runs the SMS Installer-generated executable file without installing any files. By using this method, you can see how the SMS Installer-generated executable file runs without actually installing the application. You do not have to compile the installation script before using this method.
- The run mode runs the SMS Installer-generated executable file on the reference computer. You must compile the reference script before using this method.

It is a common practice to test the file and then make any necessary modifications by changing Installation Expert options and recreating the file or by changing Script Editor actions. You can rerun either the Watch Application Wizard or the Repackage Installation Wizard without losing the changes you made with Script Editor.

SMS Installer Test Mode

With the Installation Expert test mode, you can see how the SMS Installer-generated executable file runs without actually installing the application. Before testing the installation, you must first compile the installation script by using the compile mode.

To run the SMS Installer-generated executable file in test mode, click **Test** if you are in Installation Expert. SMS Installer tests the most recent file that you compiled.

If you are in Script Editor, click **Test** on the **Build** menu. The SMS Installer-generated executable file runs, but the application is not installed on the reference computer. Only files that are copied to the \Temp directory are installed. Typically, files such as Help files and DLLs are needed by the installation.

SMS Installer Run Mode

With the Installation Expert run mode, you can test an SMS Installer-generated executable file exactly as it will run on the target computer. Before testing the installation, however, you must first compile the installation script by using the compile mode.

To test the installation in run mode, click **Run** if you are in Installation Expert. SMS Installer tests the last file that you compiled. If you are in Script Editor, click **Run** on the **Build** menu.

The run mode installs the files and makes the required registry modifications. If you are testing the installation on the reference computer that was used to create the installation, it is recommended that you remove the application that was installed during the repackaging process. This includes all files and registry modifications.

If available, use the application's Uninstall program or use **Add or Remove Programs** in Control Panel.

If you select **Run in Manual Mode** on the **Options** tab in the **Advanced Configuration** dialog box, you are prompted to specify where the installation program must place the files that you want copied into the \Windows, \System, and \Temp directories. The SMS Installer-generated executable files also include command-line options that you can use to test the installation script.

Distributing SMS Installer-generated Executable Files

You can distribute an SMS Installer-generated executable file in any of the following ways:

Use software distribution

If you plan to distribute files this way, be sure to consider the options on the **SMS** tab of the **Installation Interface** attribute, as described in the "Installation Interface Attribute" section earlier in this chapter. For more information about software distribution, see Chapter 5, "Distributing Software."

Copy the installation package to a CD

If you want to distribute software using a CD, create a single SMS Installer-generated executable file. You can include all the files within the SMS Installer-generated executable file, or if you prefer, you can place files outside the SMS Installer-generated executable file and install the uncompressed files from the CD.

Copy the installation package to floppy disks

If you want to distribute the SMS Installer-generated executable files using floppy disks, choose the **Floppy-Based Installation** option within the **Installation Interface** installer attribute, and then select the size of the floppy disks so that files of the correct size are created. When you have completed your script, click **Make Floppies** on the **File** menu, and then follow the instructions. This method may require several disks.

Post the package to the Internet or on a bulletin board system

You can place the installation package in a single file or split it into several smaller files for easier downloading.

SMS Installer-generated Executable File Compilation

The final step in creating an SMS Installer-generated executable file is to compile the script file and produce the executable file or files that contain the script and all the files that are to be included in the application. When the package is compiled, the SMS Installer-generated executable file is ready for distribution.

To compile a script, click **Compile** in the **Installation Expert** dialog box. Name the installation script, and then click **OK** to create the installation file.

SMS Installer creates the following files when it compiles a script:

***Yourapp*.exe** The installation files (including a compressed version of all the files to be installed) and the installation script.

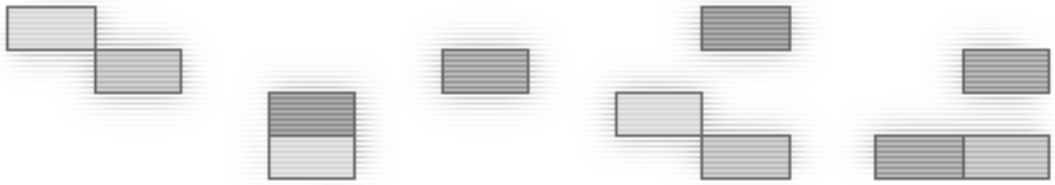
***Yourapp*.pdf** A standard SMS package definition file that is imported to distribute the SMS Installer-generated executable file to target computers with software distribution. Package definition files are created only if you select **Create Package Definition File** on the **SMS** tab in the **Installation Interface** dialog box.

***Yourapp*.ipf** The installation script, in text form.

***Yourapp*.wsm** A working file that is used by the installation script.

PART 2

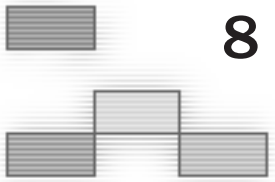
Using SMS for Change and Configuration Management



This part of the *Microsoft Systems Management Server 2003 Operations Guide* guides you through implementing Systems Management Server 2003 features in your organization.

CHAPTER 8

Software Metering



The focus of software metering in Microsoft® Systems Management Server (SMS) 2003 is the collection and reporting of software program usage data. By using software metering data, you can determine how your organization uses software programs and help ensure software license compliance. You can combine software metering program usage data with software inventory data, product compliance data, hardware inventory data, and other SMS data to create comprehensive reports.

In This Chapter

- Overview
- Configuring and Using Software Metering
- Scheduling Software Metering Maintenance Tasks
- Best Practices

For an architectural overview of software metering, see Chapter 3, "Understanding SMS Features," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Overview

SMS 2003 software metering monitors and collects software usage data on SMS clients. Data collection is based on software metering rules that are configured by the SMS administrator in the SMS Administrator console. The Software Metering Client Agent runs on the SMS client. The agent accepts software metering rules from the SMS site server and records program usage as specified in the software metering rules. Program usage data from individual SMS clients is forwarded to the client's assigned SMS site and processed by the site. The site then summarizes the data on a monthly basis and propagates the summary data to its parent site. Summarized data continues to flow up the SMS hierarchy to the central site. The central site contains program usage data from all SMS clients within the SMS hierarchy that are assigned to sites that have software metering enabled.

After you collect data from SMS clients, you can use different features to view the data, including collections, queries, and reporting. This data, combined with data from software inventory, can assist your organization in determining:

- How many copies of a particular software program have been deployed to the computers in your organization. Among those computers, you can determine how many users actually run the program.
- How many licenses of a particular software program you need to purchase when you renew your license agreement with the software vendor.
- Whether any users are still running a particular software program. If the program is not being used, you might consider retiring the program.
- Which times of the day a software program is most frequently used.

How Software Metering Works

You use software metering to monitor software program usage. Specifically, you monitor executable programs. An executable program is a compiled program that has been translated into computer code in a format that can be loaded into memory and run by the computer's processor. You specify the monitored program by the name of its executable program. Most executable programs have .exe or .com file name extension. However, SMS can monitor executable programs with other file name extensions or file names that have been renamed.

Note

The words *software program*, *executable program*, and *program* are used interchangeably in this chapter. They all refer to an executable program.

When a monitored program runs on an SMS client, software metering collects the program file information (such as file name, file version, and file size) and the program's start time and end time. For information about the data that software metering collects and reports, see the "Using Software Metering Data" section later in this chapter.

Software metering data is collected on the client when the Software Metering Client Agent is enabled. The Software Metering Client Agent examines each program that is running on the client and determines if the program matches a specified rule for the SMS site to which the client is assigned.

Usage data is collected each time a monitored program runs on the client, regardless of whether the client is connected to the network. If the client is not connected to the network, the data remains on the client and is uploaded to the SMS site server the next time that the client connects to the network and a usage upload interval has passed.

The amount of software metering data that is stored in the SMS site database is managed by an SMS process called data summarization. To improve reporting performance, SMS maintenance tasks run periodically to summarize the transactional data and delete old data, which reduces the amount of data that is retained.

Software metering reports can be integrated with SMS software inventory data that is stored in the SMS site database. When the SMS client reports program usage, it reports the same identifying information for the executable program that SMS software inventory reports. This means that software metering can report whether a particular executable program was found on a computer and whether the executable program was run on that computer during a particular time interval. For more information about collecting software inventory, see Chapter 2, "Collecting Hardware and Software Inventory."

Note

Software inventory data that is already collected by SMS can help the SMS administrator determine which executable programs to monitor with software metering. Software metering can monitor any executable program that appears in SMS software inventory.

Changes to Software Metering

Software metering has changed significantly from software metering in SMS 2.0:

- In SMS 2003, software metering uses Windows Management Instrumentation (WMI) to store software metering rules and data. This means that software metering data is stored in the SMS site database, along with other resource data that is collected by SMS. This integration of software metering with SMS makes software metering easier to use and configure in the SMS Administrator console, which contains a new **Software Metering Rules** item. Like queries for other SMS data, the software metering queries that you create are accessed from the **Query** item in the SMS Administrator console.
- SMS 2003 contains a new Web reporting tool and new software metering reports that are used to view software metering data through the tool.
- Software metering in SMS 2003 supports monitoring programs that are running in a Terminal Services session.

If you previously used SMS software metering or you are upgrading from SMS 2.0 to SMS 2003, it is important to understand the following software metering differences between these versions:

- Any data that is collected using SMS 2.0 cannot be migrated to your SMS 2003 site database.
- Software metering rules that are created in SMS 2.0 cannot be migrated to SMS 2003.
- SMS 2003 software metering sites do not recognize SMS 2.0 software metering servers.
- SMS 2003 software metering data cannot be viewed from an SMS 2.0 site and vice-versa.

In a mixed-version hierarchy, an SMS 2.0 site must be a child of an SMS 2003 site. In a mixed-version hierarchy, the SMS 2.0 software metering data flow stops at the SMS 2.0 software metering Microsoft SQL Server™ database. The data does not reach SMS 2003 sites. You can view this data only from software metering in the SMS 2.0 Administrator console tools item or through the SMS 2.0 SQL Server views (provided by the SMS 2.0 Feature Pack Web Reporting Tool).

Note

An SMS 2.0 site cannot be a parent to an SMS 2003 site. Software metering rules from an SMS 2003 site are not replicated to SMS 2.0 child sites.

Configuring and Using Software Metering

The SMS Administrator console provides basic component configuration, software metering rule specifications, and a way to display and summarize program usage data.

The following sections describe configuring and using software metering. To monitor software programs, you must:

- Enable and configure the Software Metering Client Agent.
- Create and configure software metering rules.

Enabling Software Metering

To enable software metering in SMS, you must enable and configure the Software Metering Client Agent.

To enable the Software Metering Client Agent

1. In the SMS Administrator console, navigate to **Client Agents**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
    └ Site Settings
     └ Client Agents
```

2. Click **Client Agents**.
3. In the details pane, right-click **Software Metering Client Agent**, and then click **Properties**. The **Software Metering Client Agent Properties** dialog box opens.
4. In the **Software Metering Client Agent Properties** dialog box, click the **General** tab, and then select **Enable software metering on clients**.

When you configure the agent, the changes that you make in the **Software Metering Client Agent Properties** dialog box are valid for the entire SMS site.

On the **Schedule** tab, specify how frequently you want to collect program usage data. You can also specify how often the Legacy Client downloads software metering rules from the site server. Advanced Clients download software metering rules based on the polling schedule that is configured in the Advertised Programs Client Agent. To avoid network performance problems, do not schedule downloads too frequently.

Note

The minimum recurrence interval for the data collection schedule and the metering rules download schedule is 15 minutes. If you enter an interval that shorter than 15 minutes and click **OK** on the **Schedule** tab, the recurrence time reverts to 15 minutes.

For more information about scheduling these tasks, see the SMS 2003 Administrator Help.

Excluding Advanced Clients from Software Metering

On Advanced Clients, you can exclude individual clients from software metering through the local Advanced Client policy. For more information, see Chapter 4, "Understanding SMS Clients," in the *Microsoft Systems Management Server 2003 Planning, Concepts, and Deployment Guide*. You cannot exclude Legacy Clients from software metering.

Creating Software Metering Rules

To monitor software program usage, you must create and configure software metering rules in the SMS Administrator console. Each software metering rule specifies a single software program to monitor. The software metering rule specifies several pieces of information about the program that is monitored and how the software metering rule is applied to the client.

Depending on which sections of your organization that you want to monitor software usage, you can define software metering rules for a specific SMS site or for a specific site and all of its lower level sites. SMS stores the software metering rules that you create in the SMS site database.

For the Legacy Client, the software metering rules that are applicable to the local site are compiled into a file that is replicated to the clients through the client access point (CAP).

For the Advanced Client, the software metering rules that are stored within the SMS site database are used to generate the Advanced Client policy. The policy is transmitted and published to the Advanced Client through the management point.

Table 8.1 describes the fields that must be specified for each software metering rule.

Table 8.1 Software Metering Rule Properties

Property	Description	Wildcard character	Required field
Name	The display name of the software program to be monitored. This also serves as the rule name.	Not applicable.	Yes. This information is filled in automatically if you browse to a program name.
File name	The software program's file name (such as Notepad.exe).	Not applicable.	Yes, if **Original File Name** is not specified.
Original file name	The software program's original file name, if it has since been renamed.	Not applicable.	Yes, if **File name** is not specified.

(continued)

Table 8.1 Software Metering Rule Properties *(continued)*

Property	Description	Wildcard character	Required field
Version	The version of the software program.	Use the asterisk (*) wildcard to represent a string and match on any version and use the question mark (?) wildcard to represent a character.	No. However, it is recommended that you enter the program version number, if known. Otherwise, you should leave the default wildcard symbol, which is an asterisk (*). If you leave the **Version** property blank, software metering matches the software metering rule only if the version listed in the program header file is also blank.
Language	The language of the software program.	To specify a wildcard for **Language**, choose **Any** from the list.	Yes.
Comment	SMS administrator comments, if any.	Not applicable.	No.
Site code	The SMS site code to which the software metering rule applies and whether it applies to all of its lower level sites.	Not applicable.	Yes.

Note

Some programs function as placeholders for other programs. When you define a software metering rule, be sure that you know the name of the program that ultimately runs as a process on the client computer when you run the program. For example, if you run Pbrush.exe (Paintbrush) in Microsoft Windows® XP, it launches MSpaint.exe (Paint), which is the process that appears in Task Manager. In this case, the program that you want to monitor with software metering is MSpaint.exe, not Pbrush.exe, which is an earlier version of the program.

Software Metering Rule Matching

When a program runs on the SMS client computer, the Software Metering Client Agent checks if the program matches any of the software metering rules that are defined on the client. Then, any matching rules are applied. Data is collected on the client for the rules that are applied.

Note

When you create a new software metering rule, programs matching that rule that are already running in memory on the client do not need to be restarted to be monitored by SMS. Software metering detects the programs running in memory.

A software metering rule is considered matching and is applied to a running program if all the following are applicable:

- The file name that is specified in the software metering rule matches the program file name, as displayed in Windows Explorer.

  – Or –

  The original file name that is specified in the software metering rule matches the original program file name that is stored in the executable program's header file. The header file is the file at the beginning of a program that contains definitions of data types and variables that are used by the program's functions.

- The version that is specified in the software metering rule matches the program's version in the header file. This can include wildcard characters. Note that leaving the **Version** field blank is not the equivalent of inserting a wildcard in the field. If you want software metering to match any version of the program, you must use the asterisk (*) wildcard in the **Version** field.
- The language that is specified in the software metering rule matches the language in the executable program's header file. Note that it is automatically considered a match if the software metering rule's language version is set to **Any**.

If at least one software metering rule matches a running program, SMS collects usage data for that program. Program usage data is collected only once if a duplicate software metering rule exists. For more information, see the "Software Metering Rules with the Same Name" section later in this chapter.

Scheduling Data Flow

On the **Schedule** tab in **Software Metering Client Agent Properties**, you can configure the following data flow schedules:

- Data collection
- Software metering rules download

Note

Software metering does not collect data files that are more than 90 days old. As a result, if the data file contains an end date that is more than 90 days prior to the current time, the data is rejected, status message 5614 is returned, and the data file is moved to a special folder for corrupt files.

Data collection refers to when SMS collects software metering data from clients. Software metering rules download refers to the schedule by which the Legacy Client downloads the software metering rules that are created at its site.

The **Metering rules download schedule** item, in the SMS Administrator console, applies only to Legacy Clients. To schedule downloading on the Advanced Client, navigate to **Advertised Programs Client Agent Properties** in the SMS Administrator console and configure the policy polling interval. Remember that the schedule you configure applies to all SMS features that require Advanced Client policy downloads, such as software distribution. It does not apply to software metering only.

Configuring Security Settings

Creating and configuring software metering rules requires that you configure the appropriate SMS object security credentials for the software metering rule. Applying software metering rules to SMS sites requires that you configure the appropriate site **Meter** credentials. For more information about these credentials, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Adding and Deleting Software Metering Rules

A software metering rule can be modified or deleted only in the SMS site where the rule was created. Rules that are inherited from a higher level site can be viewed in the SMS Administrator console, but not modified or deleted.

Rules are created for individual software programs only. You cannot create a single software metering rule that monitors a suite of applications. However, you can create multiple rules with the same name to perform the same service. For more information, see the "Software Metering Rules with the Same Name" section later in this chapter.

► To add a software metering rule

1. In the SMS Administrator console, navigate to **Software Metering Rules** for the site.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Software Metering Rules
```

2. Right-click **Software Metering Rules**, point to **New**, and then click **Software Metering Rule**.
3. In the **Software Metering Rule Properties** dialog box, click the **General** tab, and then enter information in the following fields:
   - **Name** (rule name)
   - **File name** and/or **Original file name**
   - **Version**

- **Language**

Note

Click **Browse** to locate the executable program, which will fill in these properties automatically.

- In the **Site code** list, select the site to which you want the software metering rule to apply.
- If you want the software metering rule to apply to the specified site and all of its lower level sites, select the **This software metering rule applies to the specified site and all its child sites** check box.

Important

The **Site code** list and the **This software metering rule applies to the specified site and all its child sites** check box are available only when first creating the rule. They cannot be modified after the rule is created and saved.

5. Click the **Security** tab, verify or change the **Class security rights** and **Instance security rights** that apply to this software metering rule.
6. Click **OK**.

To delete a software metering rule, right-click the rule in the details pane, click **Delete**, and then confirm the deletion.

Enabling and Disabling Software Metering Rules

A software metering rule can be enabled or disabled in the SMS Administrator console by right-clicking the rule, pointing to **All Tasks**, and selecting **Enable** or **Disable** from the menu. For example, you might want to stop monitoring usage of a program yet continue to run reports on the data that you have already collected. In this case, you would disable the rule. Disabling rules that you no longer need reduces the amount of network traffic that is generated by software metering. Rule status is displayed in the details pane of the SMS Administrator console. The software metering rule is disabled on the client as soon as the client downloads the changed rule.

Detaching a child site from its parent site causes the software metering rules that are created at the parent site and that are configured to apply to child sites to be disabled at the child site. However, you can re-enable these rules as well as delete them from the child site if needed.

Using Rules in Multitiered Hierarchies

A multitiered SMS hierarchy contains at least one SMS child site. When you create a software metering rule in the SMS Administrator console, you select the site to which the software metering rule applies. You also have the option of applying the software metering rule to the specified site's lower level sites or all its child sites. The software metering data that is collected on child sites is replicated up the SMS hierarchy branch to the parent sites.

At rule creation time, carefully consider whether you want the software metering rule to apply only to the selected site or to the selected site and all of its lower level sites. For example, you might want the rule to apply only to the selected site if that site is running a particular software program that the SMS clients at its lower level sites never run. After you select **This rule applies to the specified site and all its child sites** in a rule and save changes, the rule cannot be modified. Instead, you must delete the existing rule and create a new one.

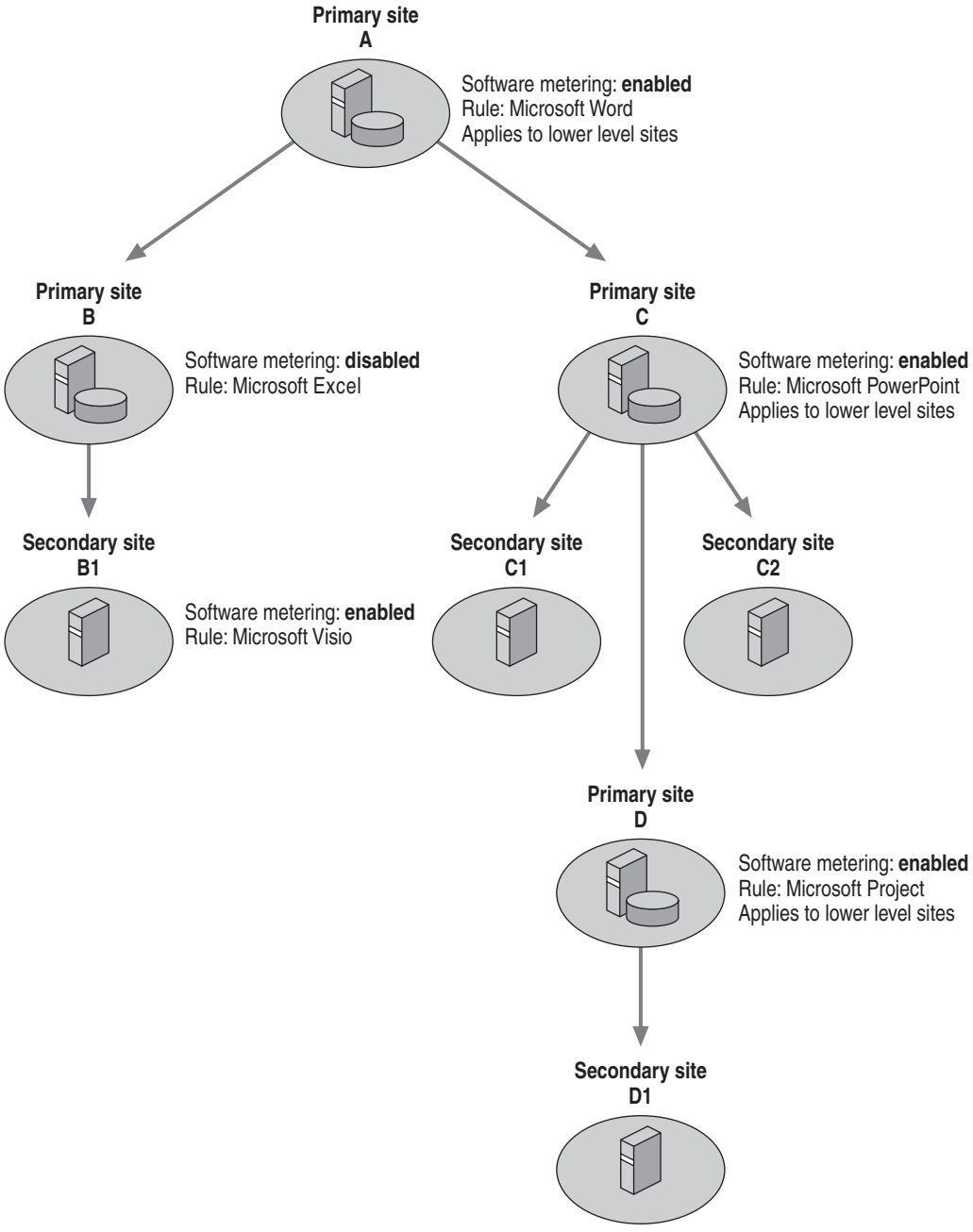
A child site receives and applies software metering rule additions, updates, and deletions from its parent site whenever a rule is created or changed. If a software metering rule is configured to apply to the specified site and all its child sites, then the next time that the software metering rules are scheduled to download on clients at the child site, the modified software metering rule is applied to those clients. Software metering rules include the site code of the site where the software metering rule was created.

When using rules in multitiered hierarchies:

- Each site in the SMS hierarchy can have its own software metering rules. Although each software metering rule is created at the primary site, you can select a different lower level site to apply the rule to when you create the rule. Or, you can create the rule on the parent site and choose whether the rule applies to all its child sites.
- If the Software Metering Client Agent is disabled in an SMS site, SMS still sends software metering rules that it received from parent sites to the lower level sites. This applies to rules that are configured to apply to the specified site and all its child sites.
- Software metering data is propagated up to the primary parent site.

Figure 8.1 shows a possible software metering rule configuration scenario in a multitiered hierarchy.

Figure 8.1 Site rules centrally configured in a multitiered hierarchy



In this scenario, the SMS administrator configures several rules for several different sites. To do this, the SMS administrator connects to primary site A in the SMS Administrator console. Then, the administrator creates the rules and configures them to apply to the specified site and all its child sites, as shown in Table 8.2. Table 8.3 describes the data that is collected at the clients based on these rules.

Table 8.2 Software Metering Rules Created at Each SMS Site

Software metering rule name	File name	Site	Rule applies to lower level sites
Microsoft Word	Winword.exe	A	Yes
Microsoft Excel	Excel.exe	B	No
Microsoft Visio®	Visio.exe	B1	No
Microsoft PowerPoint®	Powerpnt.exe	C	Yes
Microsoft Project	Project.exe	D	Yes

Table 8.3 Data Collected from SMS Clients Based on Their Assigned Site

Site	Software metering data collected from clients
Primary site A	Microsoft Word
Primary site B	None (the Software Metering Client Agent is disabled)
Secondary site B1	Microsoft Word, Microsoft Visio
Primary site C	Microsoft Word, Microsoft PowerPoint
Secondary site C1	Microsoft Word, Microsoft PowerPoint
Secondary site C2	Microsoft Word, Microsoft PowerPoint
Primary site D	Microsoft Word, Microsoft PowerPoint, Microsoft Project
Secondary site D1	Microsoft Word, Microsoft PowerPoint, Microsoft Project

Software Metering Rules with the Same Name

It is possible to create multiple software metering rules that have same rule name. If you want to monitor a suite of software programs, such as Microsoft Office applications, create multiple rules that are configured with the same rule name but different file names. This works well if you are careful about version numbers when you define the software metering rules.

Note

As a best practice, avoid making duplicate rules. Duplicate rules are rules in which every field is identical except for the rule ID.

If you configure a software metering rule in an SMS site to apply to all its child sites, the software metering rule is passed all the way down to the lowest level site in the SMS hierarchy branch, regardless of any intermediate rules with the same name that are configured to not apply to child sites. The data is collected as specified in the software metering rule at the higher level site.

Using Software Metering with Terminal Services

Terminal Services adds terminal support to Microsoft Windows NT® 4.0 Terminal Server Edition, Windows 2000 Server, and Windows Server™ 2003 family operating systems. Terminal Services is a multisession environment that provides remote access to a server desktop through thin client software that serves as a terminal emulator.

Background

In Windows 2000 Server, Terminal Services is deployed on the server in either application server or remote administration mode. In application server mode, Terminal Services delivers the Windows 2000 desktop and the most current Windows-based applications to computers that might not normally be able to run Windows. When used for remote administration, Terminal Services provides remote access for administering your server from virtually anywhere on your network.

In Windows Server 2003 family operating systems, Terminal Services technology is the basis for features that enable you to connect to remote computers and perform administrative tasks. These include Remote Desktop for Administration (formerly known as Terminal Services in remote administration mode), the Remote Desktop MMC snap-in, and Remote Desktop Connection.

Software Metering and Terminal Services

With software metering, program usage is monitored independently in each Terminal Server session. For example, if three users are logged into Terminal Server sessions, and all three are running a software program that matches an SMS software metering rule, this counts as three distinct usages of that program.

With Remote Desktop Connection (in Microsoft Windows XP), the remote desktop connection is treated as a local connection, not a Terminal Services session. This means that software metering tracks usage on the computer that is being remotely accessed, not on the host computer. Table 8.4 shows information about how the remote desktop connection is treated by software metering based on the operating system of the SMS client.

Table 8.4 Software Metering and Terminal Services Connections

Operating system	Remote connection type and mode	How software metering treats the connection
Windows NT 4.0 Terminal Server Edition	Terminal Services (application mode)	Terminal Server session
Windows 2000 Server family	Terminal Services (remote administration mode)	Terminal Server session
	Terminal Services (application mode)	Terminal Server session
Windows Server 2003 family	Terminal Services (application mode)	Terminal Server session
	Remote Desktop Administrator	Terminal Server session
Windows XP	Remote Desktop Connection	Local connection

Using Software Metering Data

This section describes the type of data that is collected by software metering, how the data is summarized, how to schedule data flow, and how to report the data. Raw usage data consists of program start and end times and information about the executable program.

Table 8.5 lists the software metering data that is collected from SMS clients.

Table 8.5 Software Metering Data

Usage information	File and program information
Start Time	File ID
End Time	File Name
Meter Data ID	File Version
Resource ID (Computer Name)	File Description
User Name	File Size (KB)
In Terminal Services Session	Company Name
Still Running	Product Name
	Product Version
	Product Language

Data Summarization

SMS clients can produce a large amount of software metering data which, when stored in its raw format, can consume a large amount of space in the SMS site database. To prevent this, background tasks run periodically to summarize the transactional data and delete old data. The data is condensed to improve reporting performance and reduce the load on your network. This data summarization reduces the amount of space that is required to store software metering data long term. Data containing greater detail is stored in the SMS site database, but for less time than summarized data.

After clients have reported software metering data for a new software metering rule, you must wait for the next summarization cycle to be completed before you can view data based on that rule. By default, the summarization site maintenance tasks run on a daily basis.

There are two types of summarized data:

Monthly usage summary data contains information about the number of times a program is run by a specific user on a specific computer.

File usage summary data contains information about the total number of distinct users for a particular software program during a specified time interval in an SMS site. This summary data is an approximation of the total number of concurrent users for the particular program being monitored. The shorter you set the recurrence interval for the data collection schedule, the less accurate this number is in approximating the number of concurrent users.

For more information about data summarization, see the "Scheduling Software Metering Maintenance Tasks" section later in this chapter.

Distinct users vs. concurrent users

The number of distinct users reported to SMS for a particular program might be higher than the number of concurrent users, but it will never be lower. This is by design. The longer that the user runs the program, the more accurate the distinct user count is (that is, the closer that number is to the number of concurrent users).

The summarization task interval is 15 minutes. For example, one user runs the program and uses it for seven minutes before closing it. Immediately afterward, another user runs the program and uses it for seven minutes before closing it. This counts as two distinct users, even though their usage does not overlap within the interval. However, if the users use the program for longer than seven minutes, the usage will overlap and the distinct user count accurately represents the number of concurrent users.

For more information about getting accurate file usage summary data, see the "Best Practices" section later in this chapter.

Software Metering Reporting

You can use SMS reporting to run a number of predefined reports for displaying information that is related to software metering. These predefined reports are grouped into the software metering category. You can also create custom software metering reports for this category.

For example, you might want to create a report that compares software inventory to actual program usage for a particular software program. This type of report can help you determine if you can reduce the number of licenses that is purchased for the program.

Some of the software metering reports that are included with SMS 2003 use software inventory data. To use these reports, you must first run software inventory on the site. For more information, see Chapter 2, "Collecting Hardware and Software Inventory."

Creating and Running Reports

You must have **Create** permission for the Reports security object class to create or import reports. You must also have the appropriate permissions for the Reports security object class or instance to modify, delete, export, or run a report. For more information about these permissions, see Chapter 5, "Understanding SMS Security" in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

The default software metering reports that show data about which software programs were run do not present useful information until software metering data has been reported by SMS clients and summarized in the SMS site database.

For information about creating and running SMS reports, see Chapter 11, "Creating Reports."

Note

Software metering reporting does not function unless you have a reporting point set up and enabled with Internet Information Services (IIS). For more information, see Chapter 15, "Deploying and Configuring SMS Sites," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide.*

Sample Reports

Several sample software metering reports are included in SMS 2003. To view these reports in the SMS Administrator console, click **Reporting**, click **Reports**, and then click **Category** in the details pane to sort the reports by category. Scroll down to the reports that are in the **Software Metering** category.

For more information about creating reports and writing queries, see Chapter 11, "Creating Reports."

Software Metering Queries

Like reports, you can create queries that are based on software metering data. Use queries to search for something particular in your SMS site database. For example, you can use software metering to locate a computer that has run a particular software program. Then, you can use this information to direct software distribution toward computers that have recently run that particular program. Or, you can use it in conjunction with the product compliance feature in evaluating compliance levels of software in your organization.

For more information about performing queries, see Chapter 4, "Managing Collections and Queries."

Scheduling Software Metering Maintenance Tasks

The four software metering tasks to include in your SMS maintenance and monitoring plan are:

- Delete Aged Software Metering Data.
- Delete Aged Software Metering Summary Data.
- Summarize Software Metering File Usage Data.
- Summarize Software Metering Monthly Usage Data.

These tasks are described in the following sections. By default, all four tasks are enabled in the SMS Administrator console. For information about configuring maintenance tasks in the SMS Administrator console, see Chapter 13, "Maintaining and Monitoring SMS Systems."

Note

You configure the scheduled start times for maintenance tasks in the SMS Administrator console. The **Latest start time** must be set to a later time than the **Start after time**. Setting these times too closely (for example, less than 60 minutes apart) might cause the task to not run properly.

Delete Data Tasks

These maintenance tasks remove old software metering data and summarized data from the SMS site database.

Delete Aged Software Metering Data

Use the Delete Aged Software Metering Data task to delete all summarized software metering data that is older than the number of days specified. Only the latest software metering data is left in the SMS site database.

By default, the task is scheduled to run every day and to delete software metering data that is older than five days. You can configure the number of days to be any number from 2 to 255.

Delete Aged Software Metering Summary Data

Use the Delete Aged Software Metering Summary Data task to delete summarized software metering summary data that is older than the number of days specified. Only the latest summarized data is kept in the SMS site database.

By default, the task is scheduled to run every Sunday and to delete software metering summary data that is older than 270 days. The maximum number of days you can configure it for is 370.

Note

If the Summarize Software Metering Data task and the Summarize Software Metering Monthly Usage Data task are not enabled, software metering data is not being summarized. In this case, when the Delete Aged Software Metering Summary Data task runs, it does not delete aged software metering data.

Summarize Software Metering Tasks

The Summarize Software Metering tasks perform the data summarization to compress the amount of data in the SMS site database, as described in the "Using Software Metering Data" section earlier of this chapter.

For the two software metering summarization tasks to succeed, software metering data that is at least 12 hours old must exist.

Data summarization runs daily and only runs against usage data that is older than 12 hours. Data summarization is required for all SMS software metering reports to display meaningful data. To understand what is contained in the most current set of summary data, you should know when summarization last occurred. A report for this (called Software metering summarization progress) is included as a sample report in SMS 2003.

Note

If all the software metering data that is reported by clients is less than 12 hours old when the summarization tasks run, then the Smsdbmon.log file contains an entry indicating that there is no data to summarize. This is likely to occur when you activate software metering for the first time. Subsequent summarization cycles operate normally.

Summarize Software Metering File Usage Data

The Summarize Software Metering File Usage Data task condenses software metering file usage data from multiple records into one general record. This record provides information about the program name, version, language, and number of distinct users over intervals of 15 minutes and one hour. This compresses the amount of data in the SMS site database.

By default, the Summarize Software Metering File Usage Data task runs daily. For every hour and every 15 minute interval within the hour, the task calculates the total number of distinct user/computer combinations that is running the matching program. Within the 15 minute intervals, this approximates the number of concurrent users. For example:

- If the same user is using a software program and is logged on to three different computers simultaneously, this counts as three usages.
- If three users are logged on to a computer running Terminal Services and all three are running the software program, this counts as three usages.
- If the same user starts and stops the software program on the same computer three separate times during the hour, this counts as one usage for that user.

When replicated up the SMS hierarchy, the software metering summary data from each site remains separated from data from the other sites. When the data reaches a parent site, each record is marked with the site code of the site where the usage data was generated. These records can be added together to estimate concurrent program usage in the network.

Summarize Software Metering Monthly Usage Data

The Summarize Software Metering Monthly Usage Data task condenses detailed software metering usage data from multiple records into one general record. This record provides information about the program name, version and language, program running times, number of usages, last usage, user name, and computer name. Data summarization helps compress the amount of data in the SMS site database. Monthly software usage data is sent to the central site.

The summarization information includes the number of times each matching software program ran on a particular computer by a particular user during the month. By default, the task is scheduled to run daily and the summarization period is one month. Software monthly usage data is replicated to the parent site.

To view software metering summarizations, you must either run queries on the summarizations or use SMS reporting. For more information about queries, see Chapter 4, "Managing Collections and Queries." For more information about the SMS reporting tool, see Chapter 11, "Creating Reports."

Best Practices

The following sections briefly describe software metering usage and configuration issues to help SMS administrators avoid common problems.

Distributing and Inventorying Programs to Be Monitored

If you want a program to be monitored by software metering, it must exist on the SMS client computer. Use SMS software inventory to determine which clients are running a particular program. If the program is not yet installed on the client, use SMS software distribution to distribute the program to clients before creating a software metering rule for that program.

Configuring a Data Collection Schedule

The default data collection schedule for the Software Metering Client Agent is every seven days. As a best practice, do not change this default setting in your production environment. If you configure data collection for a shorter time period, you begin to reduce the accuracy of software metering reporting. Also, setting this interval for a shorter time period reduces the SMS site server's ability to process data for a large number of clients. Although the minimum recurrence interval for the data collection schedule is 15 minutes, avoid configuring the interval for such a short period of time in your production environment.

Configuring Software Metering Rules

How you configure software metering rules affects metering results. The number of rules that you create can affect site system performance. The following sections describe some best practices when creating software metering rules.

Performance

Do not create an excessive number of rules for one SMS site, and avoid creating duplicate rules. Use the software metering maintenance tasks to summarize the data.

Accurate rule matching

Input only the original file name, and not the file name, in the software metering rule. This ensures that the program's usage is still monitored by SMS, even if the executable program file name has been modified on the client computer. If one of the software metering rules that is stored on the client specifies an original file name, SMS examines the header files of every program that is run on the client.

It is possible that some program header files do not contain an original file name, depending on the manufacturer. Or, the header file might have a different file name than is expected. It is good to test for these possibilities when you create software metering rules. The SMS administrator might use or devise tools to read a program header file and determine the true original file name. Otherwise, this information can be viewed manually by looking at the **Version** tab of the file properties. For more information about obtaining the original file name for a program, see your Windows documentation.

Program version issues

Executable programs contain a header file that stores the version number in two fields. One field stores the program version as a text string. The other stores the version number as a numeric value (double word or **DWORD**). SMS software inventory and software metering both use the text string value to obtain the file version of a program. They do not use the numeric value from the header file. Remember this when manually configuring the **Version** property in a software metering rule.

Also, when determining a program's version, be aware that the file version that is displayed in Windows Explorer (when you right-click a file in Windows Explorer and then click **Properties**) might not be the text version of the file. Depending on the operating system, this might be true when the program's numeric version is different from its text version.

For example, in Microsoft Windows 98 and Windows NT 4.0, the file version that is displayed in Windows Explorer is the text version. The numeric version is discarded. In Windows 2000, if the text version is not equal to the numeric version for the executable program, the file version that is displayed in Windows Explorer is the numeric version. If the file's numeric version is null or blank, the file version that is displayed in Windows Explorer is 0.0.0.0.

The same thing occurs in Windows XP and the Windows Server 2003 family when the text version does not equal the numeric version. However, by clicking **File Version** in **Other version information** on the **Version** tab in Windows Explorer, the text value is displayed.

As a best practice, use the **Browse** button when specifying the file name in the **Software Metering Rule Properties** dialog box.

For more information about obtaining version information for executable programs, see your Windows documentation.

Addressing Privacy Concerns

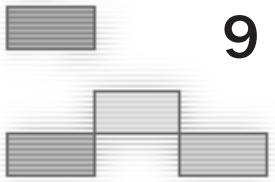
Uninformed users in your organization might be concerned that software metering is an invasion of privacy. Proactive communication can prevent this misconception.

Before implementing software metering, inform your users that you are enabling this feature. Let users know that software metering ensures software license compliance in your organization. Tell them that software metering monitors only executable programs being run on their computers, not keystrokes or work activity.

For many organizations, end-user computers are business resources that must be managed and used in a manner that is consistent with the organization's policies.

CHAPTER 9

Remote Tools



Microsoft® Systems Management Server (SMS) 2003 Remote Tools is a suite of complementary applications that you can use to access any client in an SMS hierarchy that has the Remote Tools Client Agent components installed. By using Remote Tools, you can provide assistance and troubleshooting support from your computer to clients within your site. You can use Remote Tools to access and control clients that are using the Legacy Client or the Advanced Client.

You can use Remote Tools across a wide area network (WAN) or Microsoft Remote Access Service (RAS) links to assist clients in remote locations. Remote Tools supports RAS connections with a minimum speed of 28.8 Kbps. You can also establish a connection to your organization and then access clients on your network.

In addition to SMS Remote Tools, which you can use to assist any supported client, SMS 2003 integrates Remote Assistance and Terminal Services into the SMS Administrator console for assisting applicable clients. You can also use the SMS Administrator console to manage and configure Remote Assistance settings for applicable clients on a site-wide basis.

Note

Remote Desktop Connection is the name used in Microsoft Windows® XP Professional and the Microsoft Windows Server™ 2003 family for the technology previously called Terminal Services.

Most of this chapter applies to configuring and using SMS Remote Tools. This chapter also explains how to manage, configure, and start both Remote Assistance and Terminal Services in the SMS Administrator console.

In This Chapter

- SMS Remote Tools Overview
- Remote Assistance and Terminal Services Overview
- Installing, Enabling, and Configuring SMS Remote Tools
- Configuring Site-wide Settings
- Providing Remote Support
- Advanced Features of SMS Remote Tools
- Improving the Performance of SMS Remote Tools

SMS Remote Tools Overview

The SMS Remote Tools suite consists of the following tools:

- Remote Control
- Remote Reboot
- Remote Chat
- Remote File Transfer
- Remote Execute
- SMS Client Diagnostics
- Ping Test

The following sections briefly describe each of these tools. For more information about how to use these tools, see the "Using SMS Remote Tools to Support Clients" section later in this chapter.

Remote Control

You can use Remote Control to operate a remote client. By establishing a Remote Control session, you can access the client's desktop and files and perform mouse and keyboard functions as though you were physically at the client. You can also use Remote Control to troubleshoot hardware and software configuration problems on a client and to provide remote help desk support when access to the user's computer is necessary.

Remote Reboot

You can use Remote Reboot to remotely shut down and restart a client. It might be necessary to restart a remote client to test a change to a startup procedure, to load a new configuration, or if a client is generating a hardware or software error.

Remote Chat

You can use Remote Chat to communicate with the user at a remote client. When you initiate a chat session with the user, the Remote Tools window becomes the chat window on your computer. On the remote client, a chat window also opens on the desktop. When either user types in their **Local user** box, that text also appears in the **Remote user** box on the other computer.

Remote File Transfer

You can use Remote File Transfer to copy files between the computer on which you are running the SMS Administrator console and a selected client. For example, if you discover a corrupt or missing file on a client, you can use Remote File Transfer to transfer the required file from a local file directory to the client. You can also use Remote File Transfer to transfer files, such as log files, from the client to your computer for troubleshooting.

Remote Execute

You can use Remote Execute to run executable files on a remote client. You can also run any command-line statement to complete tasks, such as running a virus checker on the client.

SMS Client Diagnostics

You can use SMS to run diagnostics on all clients. You can then use the information that is gathered to troubleshoot client hardware or software problems. For clients running Microsoft Windows NT® 4.0 or later, you can use Windows Diagnostics in the SMS Administrator console. For clients running Microsoft Windows 98, you can run diagnostics from the Remote Tools window after you have initiated a Remote Tools connection to the client.

Ping Test

You can use Ping Test to determine the reliability and speed of the Remote Tools connection to a client on your network. You can access Ping Test from the Remote Tools window.

Remote Assistance and Terminal Services Overview

The Remote Assistance and Terminal Services features, which are available in the applicable Windows operating systems of clients, are integrated into the SMS 2003 Administrator console. This provides you with more options for remotely assisting clients from within the SMS Administrator console. You can also configure and apply site-wide Remote Assistance settings for applicable clients from within the SMS Administrator console. For more information, see the "Configuring Site-wide Settings" section later in this chapter. No status messages are generated by SMS when you use Remote Assistance and Terminal Services from within the SMS Administrator console.

The Remote Assistance and Terminal Services options are dependent on the operating systems that are used for both the client and the computer from which you are running the SMS Administrator console. In some situations, both the Remote Assistance and Terminal Services options might be available for a given client.

In the SMS Administrator console, when you right-click a client in a collection and point to **All Tasks**, the **All Tasks** menu opens. The **All Tasks** menu contains the **Start Remote Tools** command, which you can use to assist any client in your site.

When both the client and the computer from which you are running the SMS Administrator console are running either Windows XP Professional or Windows Server 2003, the **Start Remote Assistance** command automatically appears on the **All Tasks** menu. You can use the **Start Remote Assistance** command to initiate a Remote Assistance session for these clients.

The **Start Remote Desktop Connection** command automatically appears on the **All Tasks** menu when the client has the Terminal Server client installed and enabled, and the client and the computer from which you are running the SMS Administrator console are both running one of the following operating systems:

- Windows NT Server 4.0, Terminal Server Edition
- Microsoft Windows 2000 Server or Windows 2000 Advanced Server

- Windows XP Professional
- Windows Server 2003 family

You can use the **Start Remote Desktop Connection** command to initiate a Terminal Services session for these clients.

The client operating system data that SMS uses to determine the availability of Remote Assistance and Terminal Services is based on discovery data. Some discovery methods, such as Network Discovery, might not provide the operating system name and version. The **Start Remote Assistance** and **Start Remote Desktop Connection** commands might not appear until an SMS client is installed and a discovery data record is generated.

Notes

- The appearance of commands on the **All Tasks** menu indicates only the possibility of the client to be controlled, it does not indicate that the feature is installed and enabled on the client.
- On computers running Windows 2000, installing the SMS Administrator console upgrades the Terminal Services client to the Windows Server 2003 version of the Remote Desktop Connection application.

To start a Remote Assistance or Terminal Services session by using the SMS Administrator console

1. In the SMS Administrator console, navigate to **Collections**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ Collections
    └ collection containing client
```

2. Locate a collection that contains the client with which you want to start a session.
3. Right-click the client, point to **All Tasks**, and then click **Start Remote Assistance** or **Start Remote Desktop Connection**.

Note

When you initiate a Remote Assistance session in the SMS Administrator console, Remote Assistance cannot automatically detect the speed of the network connection to the client. The session always assumes that a slow network connection exists. This provides the fastest possible performance in all situations.

For more information about using Remote Assistance and Terminal Services to control and assist clients, see the Windows operating system documentation.

Installing, Enabling, and Configuring SMS Remote Tools

SMS Remote Tools requires installing and configuring components on both the SMS site server and the clients. If you select the Remote Tools option in the setup wizard, the Remote Tools server components are installed during a primary or secondary site installation, or during an SMS Administrator console installation.

Before you can use Remote Tools to connect to and support clients, you must enable and configure the Remote Tools Client Agent settings for the site. After you enable Remote Tools on a site, the Remote Tools Client Agent components are installed when new clients are installed to that site, or when clients that are already installed update their site configuration.

Enabling and Configuring the SMS Remote Tools Client Agent on the SMS Site Server

You use the SMS Administrator console to enable and configure the Remote Tools Client Agent settings. The settings that you specify for each site apply to all the clients that are assigned to that site.

Important

Before enabling SMS Remote Tools for a site, see the "Configuring Site-wide Settings" section later in this chapter to determine which Remote Tools Client Agent settings are relevant to your site. Pay special attention to the settings on the **Advanced** tab, because these settings are difficult to change after the Remote Tools Client Agent components have been installed on clients.

After you have installed the SMS primary site and verified that all SMS services are running correctly, you can enable Remote Tools on the site.

To enable Remote Tools on the SMS site server

1. In the SMS Administrator console, navigate to **Client Agents**.

```
Systems Management Server
 └ Site Database <site code - site name>
  └ Site Hierarchy
   └ <site code - site name>
   └ Site Settings
    └ Client Agents
```

2. In the details pane, right-click **Remote Tools Client Agent**, and then click **Properties**.

3. In the **Remote Tools Client Agent Properties** dialog box, click the **General** tab, and then select the **Enable remote tools on clients** check box.

Installing SMS Remote Tools on Clients

The Remote Tools Client Agent components are not fully installed on clients until after you enable Remote Tools on the SMS site server. You must also enable and initiate client discovery and installation methods on the site server. For more information about client discovery and installation methods, see Chapter 4, "Understanding SMS Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Remote Tools Installation on Legacy Clients

After you enable Remote Tools on the site server, when Legacy Clients are installed on the site, the Remote Tools Client Agent components are automatically installed on each client. This occurs when the Client Component Installation Manager (CCIM) checks its client access point (CAP), discovers that Remote Tools has been enabled, and installs the necessary components.

The CCIM is an SMS client component that ensures that each Legacy Client is properly installed and assigned to the correct site. The CCIM also keeps the client data and the SMS site server data synchronized by creating discovery data records, and it determines which optional components should be installed. This component runs as a thread of the SMS Client service.

After the Remote Tools Client Agent components are installed on a Legacy Client, you have full Remote Tools functionality, with the following exceptions:

- Clients running Windows NT 4.0 require a restart to load low-level drivers, as described in the "Installation on Clients Running Windows NT 4.0" section later in this chapter.
- Clients running Windows 98 require a restart to enable full-screen MS-DOS® sessions and some keyboard features, as described in the "Installation on Clients Running Windows 98" section later in this chapter.

Remote Tools Installation on Advanced Clients

After you enable Remote Tools on the site server, when Advanced Clients are installed on the site, the Remote Tools Client Agent components are automatically installed on each client. However, you can prevent the installation of the Remote Tools component by selecting the **Do not install Remote Control components for Advanced Clients running Windows XP, Windows Server 2003, or later** check box. The installation of the Remote Tools component occurs when the Client Configuration Manager (CCM) Policy Agent checks its management point and discovers that Remote Tools has been enabled and the Remote Tools Client Agent installs the necessary components.

When installing an Advanced Client, you have the option of installing the Remote Tools components at the same time, instead of waiting for the site server to pass Remote Tools policy down to the client. You can do this by using the following command-line setup option.

```
Msiexec /i Client.msi SMSFULLREMOTETOOLS=1
```

This sets up the Remote Tools Client Agent components on the client with default Remote Tools configuration settings.

> **Note**
> Before using this option, ensure that Remote Tools is enabled for the site. Otherwise, the Remote Tools Client Agent components are disabled when the client contacts the management point.

For more information about installing clients, see Chapter 4, "Understanding SMS Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Installation on Clients Running Windows 2000 or Later

SMS 2003 provides full Remote Tools support for clients running Windows 2000 or later, both in Windows domains and in native mode or mixed mode Active Directory® domains.

On clients running Windows 2000 or later, SMS installs a virtual keyboard and mouse driver named KBSTUFF.sys. This driver functions as both the SMS Virtual Keyboard and the SMS Virtual Mouse. Because clients running Windows 2000 or later have a Plug and Play driver model, it is not necessary to restart the client after installation to have full Remote Tools functionality. To uninstall Remote Tools from a client running Windows XP, it is necessary to restart the client.

Installation on Clients Running Windows NT 4.0

To ensure full Remote Tools functionality on clients running Windows NT 4.0, you must restart the clients after you install the Remote Tools Client Agent components. On clients running Windows NT 4.0, the Remote Tools Client Agent relies on two low-level drivers: KBSTUFF.sys and RCHELP.sys. KBSTUFF.sys emulates a keyboard and some custom-pointing devices on the client. If it is not properly installed, keyboard and mouse drivers do not function properly. RCHELP.sys determines video driver compatibility.

It is important to note that a restart is also required to uninstall these drivers from a client running Windows NT 4.0. This is especially important if you enable and disable the Remote Tools Client Agent for an SMS site multiple times. Because a client running Windows NT 4.0 requires a restart to install the low-level drivers, it is common for a subsequent installation of these components to fail due to a previous incomplete installation. For example, if the Remote Tools Client Agent is installed on a client running Windows NT 4.0, but the client is not restarted, the low-level drivers are not completely installed. If the administrator disables the Remote Tools Client Agent on this site before the client is restarted, the client components are flagged for deletion during the next client restart, but they still remain installed. Any subsequent installation attempt fails because the incoming drivers cannot overwrite the existing versions. If these drivers fail to install, check the Remctrl.log file to determine whether the drivers were successfully installed previously. The Remctrl.log file is located in the %SystemRoot%\MS\SMS\Logs directory on the client.

Preinstallation Testing for Clients Running Windows NT 4.0 or Later

Before installing the Remote Tools Client Agent components on clients running Windows NT 4.0 or later, you should perform lab testing to identify the following potential problems:

- Video driver compatibility on clients running Windows NT 4.0
- Conflicts with third-party client agents on clients running Windows NT 4.0 or later

Video Driver Compatibility

Video acceleration significantly speeds up your Remote Control sessions with clients. For video acceleration on clients running Windows 2000 or later, SMS uses a Mirror driver. The Mirror driver can simultaneously display the same output to several video devices and has no dependencies on the client's video driver.

Clients running Windows NT 4.0 might have problems with video driver compatibility. Before you use video acceleration on clients running Windows NT 4.0, you should:

- Test the compatibility of the accelerator driver with the client's video driver. For more information, see the "Video Acceleration" section later in this chapter.
- Ensure that the video drivers on your clients are on the list of tested and supported video drivers. For more information, see the "Video Drivers That Can Be Accelerated for Clients Running Windows NT 4.0" section later in this chapter.

Conflicts with Third-party Client Agents

The SMS Remote Control Agent can conflict with third-party remote control applications that use the same executable file name (Wuser32.exe). The Remote Tools Client Agent installation program for the Legacy Client determines if any conflicting remote control agents are on the client before installing the Remote Tools Client Agent components. If conflicting agents are present, the components are not installed. The Remote Tools Client Agent installation program does not perform this check on the Advanced Client. For either the Advanced or Legacy Client, if conflicting third party products do exist on the computers, then you should remove the conflicting products, or you should not enable Remote Tools for that SMS site.

You can check the installation status by using **System Management**. On the client, open Control Panel, double-click **System Management**, and then click **Components**. If the agent failed to install, the Remote Control Agent value is set to **Not Available**.

When the Remote Tools Client Agent components cannot be installed, the CCIM generates a status message. The status message is sent to the SMS site to alert the administrator that the client agent failed to install. Although the status message does not contain the reason for the failure, the Remctrl.log file on the client does contain this information.

On the Legacy Client, the Remctrl.log file is located in the following directory:

%SystemRoot%\MS\SMS\Logs

On the Advanced Client, the Remctrl.log file is located in the following directory:

%Windir%\system32\CCM\Logs

For the Legacy Client, the CCIM attempts to install components that are set to **Not Available** every 30 days. If the conflicting third-party agent has been removed, the Remote Tools Client Agent components are installed.

For both the Legacy Client and the Advanced Client, you can manually attempt to install the Remote Tools Client Agent components. To do this, open Control Panel on the client, double-click **Systems Management**, and then click **Repair Installation**. If no conflicting remote control agents are found, the Remote Tools Client Agent components are installed.

You can enable additional logs for tracking Wuser32.exe on a client computer, and for the Remote Control Client Viewer on the computer running the SMS Administrator console. To enable logging for Wuser32.exe, set the value of **LogToFile** to **1** in the client's registry under **\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SMS \Client\ Client Components\Remote Control**.
The resulting log file is named Wuser32.log.

On the Legacy Client, the Wuser32.log file is located in the following directory:

%SystemRoot%\MS\SMS\Logs

On the Advanced Client, the Wuser32.log file is located in the following directory:

%Windir%\system32\CCM\Logs

To enable logging for the Remote Control Client Viewer on the computer running the SMS Administrator console, set the value of **LogToFile** to **1** in the registry under **\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SMS \Components\SightNT\Viewer**. The resulting log file is named Remote.log and the file is located in the SMS\bin folder on the SMS site server or the computer running the SMS Administrator console.

Installation on Clients Running Windows 98

For clients running Windows 98, the virtual device driver (V*x*D) is inserted into the Windows registry to load the Vuser9x.vxd driver. Until the client is restarted, Vuser9x.vxd cannot be loaded. Without this driver, full-screen MS-DOS sessions and some keyboard features do not work correctly during a Remote Control session.

Confirming SMS Remote Tools Installation

To confirm that the Remote Tools Client Agent components have been installed on a client, verify that there is a *.log file on the client as follows:

- Legacy Client (*%SystemRoot%*\MS\SMS\Clicomp\RemCtrl\Install.log)
- Advanced Client where Ccmsetup.exe is used to install the client (*%SystemRoot%*\System32\CCMSetup\Client.MSI.log)

The install *.log file contains a list of the installation tasks that ran during the installation or removal of the Remote Tools Client Agent components, including registry key creation or removal.

You can also view the Remctrl.log file at the following directory on the client:

- Legacy Client (*%SystemRoot%*\MS\SMS\Logs)
- Advanced Client (*%SystemRoot%*\System32\CCM\Logs)

The Remctrl.log file is more detailed and records all significant actions that the Remote Tools Client Agent performs. The Remctrl.log file is essential for identifying Remote Tools functions after the Remote Tools Client Agent components are installed and running. It is also essential for identifying Hardware Munger and Security Munger actions. The Remctrl.log file does not provide information about Remote Control session functions.

The Remctrl.log file provides detailed information about:

- Operating system and local client language settings.
- Actions performed by the Hardware Munger and the Security Munger on the Legacy Client.
- Actions performed by the Remote Tools Client Agent on the Advanced Client.
- Installation and removal of the Remote Tools Client Agent components.

Configuring Site-wide Settings

You use the **Remote Tools Client Agent Properties** dialog box to configure your site settings. The tabs contain properties that you can set to customize Remote Tools for the clients on your site. For example, you can specify whether client users must grant permission before an administrator can conduct a Remote Control session, the level of security, and protocol-related settings. These settings apply to all clients in your site.

You can also manage and configure Remote Assistance settings that apply to all applicable clients in your site. If you choose to manage Remote Assistance settings by using SMS, you can override user Remote Assistance settings and choose the level of Remote Assistance available to administrators.

The tabs included in this dialog box are:

- **General**
- **Security**
- **Policy**
- **Notification**
- **Advanced**

General Tab

The **General** tab contains settings that apply to both SMS Remote Tools and Remote Assistance. You can use this tab to:

- Enable Remote Tools for all clients within the site.
- Prevent client users from changing **Policy** or **Notification** tab settings.
- Choose whether to manage Remote Assistance settings for applicable clients within the site and whether to override Remote Assistance user settings.

The **Users cannot change Policy or Notification settings for SMS Remote Tools** check box is cleared by default. If you select this check box, it means that all clients in the site must use the settings that you specify for the site. Users cannot change the local Remote Tools settings on clients. If you do not select this check box, users can change the following Remote Tools options:

- The Remote Tools functions that an SMS administrator can perform
- Whether an SMS administrator must ask permission before a Remote Tools session can be established
- Whether visual or audio indicators announce that a Remote Control session is taking place
- Whether to display the Remote Tools taskbar indicator in the notification area or as a high-security indicator on the client desktop
- Whether the Remote Control components are installed on Advanced Clients running Windows XP Professional or Windows 2003 Server

Select the option **Do not install Remote Control components for Advanced Clients running Window XP, Windows Server 2003, or later** to prevent Remote Control from being installed on computers running those platforms. It is strongly recommended that you use the Windows Remote Assistance and Remote Desktop Connection features of Windows XP and Windows Server 2003 rather than SMS Remote Control on computers running those platforms. Windows Remote Assistance and Remote Desktop Connection are more secure technologies and are built-in features of the operating system.

Security Tab

The **Security** tab contains settings that apply both to SMS Remote Tools and to Remote Assistance.

The **Permitted Viewers** list applies to both SMS Remote Tools and Remote Assistance users. You can use this tab to add non-administrators users and user groups to the **Permitted Viewers** list. Permitted viewers are users and user groups that can remotely access clients running Windows NT 4.0 or later. By using SMS 2003, members of the local Administrators group can access clients, regardless of whether they appear in the **Permitted Viewers** list.

Although the **Permitted Viewers** list appears to accept only user groups, you can also add user names to this list. It is more efficient to manage this list by using user groups, but the ability to specify a user name is available to those who need it.

When you upgrade from SMS 2.0, remove all unnecessary language-specific administrator names from the **Permitted Viewers** list. Doing so enhances the performance of SMS Remote Tools by reducing the number of permitted viewers that are authenticated by the domain controller each time you initiate a Remote Tools function. SMS 2003 Remote Tools automatically grant Remote Tools access to the Administrators group. You do not need to add the Administrators group to the **Permitted Viewers** list.

Using Remote Tools on clients running Windows NT 4.0 or later requires that the user be a member of the local Administrators group or be included in the Permitted Viewers list. For all clients, you must also create a security right to use Remote Tools on specific collections and assign that right to specific users or user groups. For more information about Remote Tools security, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Policy Tab

The **Policy** tab contains settings that apply to both SMS Remote Tools and Remote Assistance.

You can use this tab to:

- Specify the level of SMS Remote Tools access (**Full**, **Limited**, or **None**).
- Specify whether users must grant permission when an administrator tries to remotely access their client.

> **Note**
> You can limit the requirement for users to grant permission to only clients running Windows 98. This provides greater security for those clients.

- Specify the level of Remote Assistance access (**Full control**, **Limited viewing**, or **None**).

Level of SMS Remote Tools access

You can choose to allow administrators to perform all Remote Tools functions, no Remote Tools functions, or limited Remote Tools functions. If you allow administrators limited Remote Tools functions, you can then specify which functions are permitted.

To specify limited permissions

1. In the **Level of remote access allowed** list, click **Limited**, and then click **Settings**.
2. In the **Default Limited SMS Remote Tools Settings** dialog box, select the Remote Tools functions that you want administrators to have for clients of the site. For more information about these functions, see the "SMS Remote Tools Overview" section earlier in this chapter.

Level of permission required for SMS Remote Tools

You can choose to allow administrators to perform Remote Tools functions with or without client permission.

When you select the **Do not ask permission** check box, using SMS Remote Tools on clients running Windows 98 is less secure than on clients running Windows NT 4.0 or later. Specifically, there is a greater risk of an unauthorized Remote Control session to a client running Windows 98. For this reason, it is recommended that you always display a message to ask for the user's permission on clients running Windows 98. You can do this in two ways:

- Select the **Display a message to ask for permission** option, which displays a message on all clients.
- Select the **Display a message to ask for permission** option, and then select the **Only on clients running Windows 98** check box, which displays a message only on clients running Windows 98.

Level of Remote Assistance access

You can choose to allow administrators to use Remote Assistance to fully control applicable clients, to remotely view applicable clients, or to not use Remote Assistance. The level of control that you choose for this setting applies to all Remote Assistance sessions, whether you start them from within the SMS Administrator console or from the operating system.

To enable all site-wide settings for Remote Assistance on the clients, SMS passes the settings to the clients and applies them by using local Group Policy. If you subsequently apply Group Policy settings at the site, domain, or organizational unit level by using the Group Policy Microsoft Management Console (MMC) snap-in, the local Group Policy settings applied by SMS on clients are overwritten.

If you select the **Users cannot change Policy or Notification settings for SMS Remote Tools** check box on the **General** tab, the user cannot override these settings on a client.

User permission is always required when using Remote Assistance in the SMS Administrator console.

Notification Tab

The settings on the **Notification** tab apply only to SMS Remote Tools.

Note

Your organization's internal policy and, in some circumstances, the privacy laws in your locale might influence the level of user alerts that you specify.

You can use this tab to:

- Specify whether to display a visual indicator to notify users when a Remote Control session is active on their computers. This visual indicator pertains to Remote Control only, not to other Remote Tools functions.
- Select the type of visual indicator to be displayed. The visual indicators differ in where they appear on the desktop and whether the indicator can be hidden from the user's view.
- Specify whether to display the visual indicator only when a Remote Control session is active or when no session is active.

- Specify whether to play a sound to notify users when a Remote Control session is active. You can specify that the sound play only when a session begins and ends or plays repeatedly during a session.

Status indicators

There are two types of visual indicators:

Taskbar indicator The taskbar indicator appears in the notification area on the client's taskbar. The indicator changes its appearance when an SMS administrator initiates a Remote Control session with the client. You can configure the Remote Tools Client Agent to permit the user to hide this indicator.

High-security indicator The high-security indicator initially appears in the top right corner of the client's desktop. The user can move the icon but cannot hide it, which allows a user to always determine if and when a Remote Control session has been initiated. The indicator is displayed within the icon. The title bar of this indicator is gray until a Remote Control session is initiated, and then the title bar turns red.

Table 9.1 Remote Control Indicators

Icon	Description
	Taskbar indicator. No Remote Control session is active.
	Taskbar indicator. A Remote Control session is active.
	Taskbar indicator. A Remote Control session is active but paused.
	High-security indicator. No Remote Control session is active and the title bar is gray.
	High-security indicator. A Remote Control session is active and the title bar is red.
	High-security indicator. A Remote Control session is active but paused.

Advanced Tab

The settings on the **Advanced** tab apply only to SMS Remote Tools. The **Advanced** tab in the **Remote Tools Client Agent Properties** dialog box contains a number of hardware-related settings. For most installations, the default settings in this dialog box should not be changed. For more information, see the "Client Hardware Settings" section later in this chapter.

You can use this tab to:

- Select the default video compression level of remote screen captures during a Remote Control session (**Low**, **High**, or **Automatically Select**). For more information, see the "Video Compression" section later in this chapter.

- Select the default remote access protocol for all clients in the site. If you are using the SMS 2003 Administrator console to configure an SMS 2.0 site, you can select TCP/IP or NetBIOS. For SMS 2003 sites, the only supported protocol is TCP/IP and the default remote access protocol setting is not available.
- Enable video acceleration clients running Windows NT 4.0 or later and determine which video drivers can be accelerated for clients running Windows NT 4.0. For more information, see the "Video Acceleration" section later in this chapter.

Important

If you change the settings on the **Advanced** tab after the Remote Tools Client Agent components have been installed on clients, the previously installed clients do not receive the new settings automatically. The revised **Advanced** tab settings are passed down to the clients during the next maintenance cycle of the CCIM, but they are not implemented until you uninstall and reinstall the Remote Tools Client Agent components. This applies to Legacy Clients only. For more information, see the "Client Hardware Settings" section later in this chapter.

Providing Remote Support

Remote client support extends your ability to improve and maintain the operating health of the hardware and software throughout an SMS site. SMS Remote Tools, along with the integration of Remote Assistance and Remote Desktop Connection, increases the effect that you can have in supporting clients and users that are separated by time or distance. By providing remote support to clients and users, you can perform a variety of activities to solve network operations and management problems.

This section applies primarily to the usage of SMS Remote Tools to control clients. For more information about using Remote Assistance and Remote Desktop Connection to control clients, see the Microsoft Windows product documentation.

Using SMS Remote Tools to Support Clients

You can use SMS Remote Tools to perform a variety of troubleshooting activities directly from your computer to support clients in remote locations.

After you have established a Remote Tools connection, you can:

- Control clients remotely.
- Conduct two-way conversations with client users.
- Diagnose client hardware and software problems.
- Test network connectivity.
- Run commands and programs on clients.

- Transfer files to or from clients.
- Restart clients.

> **Note**
> If the site has limited the permissions to use Remote Tools, or if the user has limited the permissions to use Remote Tools on a specific client, the buttons for any restricted Remote Tools are unavailable in the **Remote Tools** window.

Establishing an SMS Remote Tools Connection

Before you can use SMS Remote Tools, you must establish a connection with the client. There are two ways to establish a Remote Tools connection:

- By using the SMS Administrator console
- By running Remote.exe directly from the command line

In the SMS Administrator console, you can establish Remote Tools connections with up to four different clients at a time. You cannot establish more than one Remote Tools connection to any one client at a time. For example, you might control two clients remotely at the same time or control one client remotely, while transferring files to another client.

To establish a Remote Tools connection, you must have **Use Remote Tools** and **Read** permissions for the collection that contains the client. If you are not a local administrator, you must also be included in the **Permitted Viewers** list, which is on the **Security** tab in the **Remote Tools Client Agent Properties** dialog box. For clients outside the SMS site boundaries or authenticating domain, correct security credentials must be provided before you can establish a Remote Tools connection to those clients. For more information about Remote Tools security, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Establishing a Remote Tools Connection by Using the SMS Administrator Console

You can establish a Remote Tools connection to a client in the SMS Administrator console.

► To establish a Remote Tools connection in the SMS Administrator console

1. In the SMS Administrator console, navigate to **Collections**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ Collections
  └ collection containing client
```

2. Locate a collection that contains the client to which you want to connect.
3. Right-click the client, point to **All Tasks**, and then click **Start Remote Tools**.

For more information about using the Remote Tools window, see the "Using SMS Remote Tools to Support Clients" section later in this chapter.

If you cannot establish a Remote Tools connection to the client, ensure that Remote Tools is enabled on the SMS site server and that the Remote Tools Client Agent is successfully installed on the client. Also, ensure that you have **Use Remote Tools** security credentials to the collection containing the selected client.

Establishing a Remote Tools Connection by Using Remote.exe

All Remote Tools functions are also available by running the Remote.exe program directly from the command line to establish a Remote Tools connection. This is useful if you are developing applications that require SMS Remote Tools functionality.

This program is located in the %*SystemRoot*%\SMS\Bin\I386 directory for a primary or secondary site installation, and in the %*SystemRoot*%\SMSADMIN\Bin\I386 directory for an SMS Administrator console installation.

Remote.exe uses the following syntax:

Remote <*Protocol_Type*> <*Address*> \\<*Site Server Name*>\ [/SMS:NOSQL]

Where:

- *Protocol_Type* is 1 for IPX, 2 for TCP/IP, or 3 for NetBIOS.

  Note
  A value of 0 introduces a special case, described later in this section.

- *Address* is a valid IPX network number, IP address or client name, or NetBIOS name.

  Examples:

  C:\SMS\BIN\I386> REMOTE 2 172.16.0.0 \\BIG_SERVER\

  C:\SMS\BIN\I386> REMOTE 3 DUBN_NETBIOS \\BIG_SERVER\

  Note
  The Internetwork Packet Exchange (IPX) and NetBIOS protocol types apply only when you conduct remote sessions on SMS 2.0 clients. SMS 2003 clients use only TCP/IP.

- *Site Server Name* is the site server name of the site to which the client belongs.

When you use Remote.exe with an explicit Protocol_Type of 2 (TCP/IP), SMS resolves a client name to its IP address and then uses that address to attempt a connection. Name resolution is not attempted when you use Remote.exe with an explicit Protocol_Type of 1 (IPX) or 3 (NetBIOS).

When you use the following syntax: **Remote 0 <***Resource_ID***>** or **Remote** (with no options), Remote.exe attempts a connection for all available protocols. To determine a client' Resource ID number, right-click a client in the SMS Administrator console under **Collections**, and then click **Properties**. The **Resource ID** field for the client appears in the <***Client***> **Properties** dialog box. You can also obtain a client's resource ID by using a custom query run through Windows Management Instrumentation (WMI).

To connect to a client by using its resource ID, use the following command syntax:

Remote 0 <*Resource_ID***> \\<***Site Server Name***>**

Example:

C:\SMS\BIN\I386> REMOTE 0 2 \\BIG_SERVER\

When you use 0 in the first parameter, Remote.exe attempts to connect by using all available protocols for the target client.

The *Site Server Name* parameter is the site server name for the site to which the client belongs.

The SMS:NOSQL option is used in place of the *Site Server Name* option to allow direct connection to the client without using data in the SMS site database. This is useful if the client's name resolution is not current, or if the client's IP address is not updated in the SMS site database. An address type of 0 is not valid when used in conjunction with the SMS:NOSQL option.

Example:

C:\SMS\BIN\I386> REMOTE 2 172.16.0.0 /SMS:NOSQL

If you use Remote.exe with no command-line options, the **Remote Tools Address Connection** dialog box appears. You can use this dialog box to enter the following parameters:

- **Address type** (**NetBIOS name**, **IP address**, or **IPX address**)
- **Address** (any valid NetBIOS name, IP address or client name, or IPX network number)

When you have entered the parameters, click **OK** to connect to the client. A connection to the client is established if the following conditions are met:

- The Remote Control Agent (Wuser32.exe) is running on the client
- The SMS Administrator console and client share a common protocol

Note

SMS 2003 Remote Control clients listen only for TCP connection attempts. NetBIOS and IPX connections are made by Remote.exe for backward capability with SMS 2.0 clients.

After a Remote Tools connection to the client is established, you can perform any of the Remote Tools functions on the client. For more information, see the "Using SMS Remote Tools to Support Clients" section earlier in this chapter.

Remotely Controlling Clients by Using SMS Remote Tools

After you successfully connect to a client by using SMS Remote Tools, you can initiate a Remote Control session. During a Remote Control session, you can take control of a client by displaying a duplicate view of the client's desktop in a window on your desktop. You can then control the client by using your keyboard and mouse. If a user is at the client, the user can still use the local keyboard and mouse, so that you can work with the user interactively.

To start a Remote Control session, establish a Remote Tools connection. Then, in the Remote Tools window, click **Remote Control**.

> **Note**
> You cannot use an SMS Remote Control session and a Remote Desktop session simultaneously to control a client running Windows XP Professional. For more information, see article 304591 in the Microsoft Knowledge Base at http://support.microsoft.com.

After you have established a Remote Control session, the client's desktop appears on your screen in the Remote Control Client Viewer window, surrounded by a moving black and yellow border. Depending on how you have configured the Remote Tools Client Agent properties for the site, you might need the client user's permission to conduct the Remote Control session.

> **Note**
> A visual indicator appears either in the notification area or on the desktop of the client to alert the user that a Remote Control session is in progress.

In addition to controlling the client by using your keyboard and mouse, you can also use the command buttons in the upper-right corner of the Remote Control Client Viewer window to perform functions, such as simulating the ALT+TAB key sequence or opening the **Start** menu on the client. For more information about using the Remote Control Client Viewer window, see the SMS Help.

> **Note**
> When you start a Remote Control session, if the NUM LOCK key settings are different on the client and on the SMS Administrator console computer, you cannot change the NUM LOCK key settings of the client by using the SMS Administrator console keyboard. You can still enter numbers on the client by using the number keys at the top of the SMS Administrator console keyboard.

A Remote Control session can be helpful for resolving a problem that a user is experiencing. By initiating a Remote Control session, you can directly view the client desktop while the user demonstrates the problem. Often, watching the user attempt a task offers useful insight into specific errors that the user is making or reveals important details about the problem. Or, from your SMS Administrator console, you can demonstrate how to complete a task correctly by performing mouse actions and keystrokes while the user watches. With Remote Control, you can also view error messages exactly as they appear on the user's screen, instead of depending on the user to paraphrase the error message.

If a user has problems completing a task, you can establish a Remote Control session and conduct an individualized training session with the user. You can also conduct a session with a problem client, establish a second session with a client that works correctly, and then compare the registry settings or the results of running a file on the two clients.

A Remote Control session can be conducted without a user being logged on to the client, because the Remote Control Agent (Wuser32.exe) remains installed and running on clients. For more information, see the "Role of Wuser32.exe on Clients" section later in this chapter.

Conducting Two-Way Conversations with Client Users

You might want to establish an on-screen conversation to communicate with a user that is logged on to a client. To begin the conversation, establish a Remote Tools connection. Then, in the Remote Tools window, click **Remote Chat**.

When you have successfully established a chat session, a Remote Chat window appears on both the administrator and client screens. Each window has two text boxes, one for the remote user and one for the administrator.

When the user at the client types in the **Local** box, the text appears in the **Remote** box on the administrator's screen. You can then respond by typing in the **Local** box, which appears in the **Remote** box on the client. This feature is especially useful when you cannot talk to the user by phone while providing them with remote support.

Diagnosing Client Hardware and Software Problems

If a user reports a hardware or software problem, you can obtain diagnostic information for clients. By using Remote Tools Diagnostics, you can obtain information such as:

- Free, used, and virtual memory.
- IRQ assignments.
- Loaded device drivers.
- Environmental variables.

Depending on the type of problem that is reported by the user, you might need to view client memory information or to know the current operational state of the client, such as free disk space. Or, you might suspect network connectivity problems. You can use the diagnostic information that you obtain to troubleshoot client hardware and software problems.

Diagnosing clients running Windows NT 4.0 or later

You can run Windows Diagnostics from the SMS Administrator console. To run Windows Diagnostics, navigate to a collection that contains the client, right-click the client, point to **All Tasks,** and then click **Start Windows Diagnostics**. The Windows Diagnostics for the client appears in a separate Systems Information console. For more information about running Windows Diagnostics, see the Microsoft Windows product documentation.

Diagnosing clients running Windows 98

For clients running Windows 98, you can run Remote Tools Diagnostics from the Remote Tools window. For more information about using Remote Tools Diagnostics from the Remote Tools window, see the SMS Help.

Testing Network Connectivity

You can use the Ping Test tool to test the reliability and speed of a Remote Tools connection and to test client connectivity with any network protocol. To use Ping Test, establish a Remote Tools connection, and then click **Ping Test**.

Ping Test sends packets to the client by using your site's default protocol. To test the connection, Ping Test sends a burst of packets to the client for four seconds. Ping Test then analyzes the number of packets that are returned by the client and the elapsed time to determine the reliability and speed of the communications channel to the client. The left side of the Ping Test window shows the speed and quality of the connection. The color red indicates poor connectivity. As the connection reliability improves, the color changes to yellow and then to green. The **Test statistics** area displays the total number of packets sent during the test, the packets returned per second, and the total errors. By using this information, you can determine the relative speed of the connection to the client.

The Ping Test tool is not the same as the Ping Provider tool that is provided in Network Trace, which uses only TCP/IP. Ping Test can test the quality of network connectivity regardless of the default network protocol that is being used.

Note

When you use Ping Test to evaluate the communication channel between the SMS Administrator console and the client, be aware that you use most of the available bandwidth of that channel for a few seconds. Depending on the network route between you and the client, performance can be affected while the connection is evaluated.

Running Commands and Programs on Remote Clients

The primary purpose of Remote Execute is to provide administrators with the ability to run applications in their own security context. Remote Control launches applications in the user's security context.

You can use Remote Execute to run any command-line statement on a remote client. To use Remote Execute, establish a Remote Tools connection, and then initiate the tool by clicking **Remote Execute**.

In the **Remote Execute** dialog box, type the name of the program or batch file that you want to run on the client. When you run a command-line statement from the Remote Execute window, the executable file must reside in the client's path. If it does not, you must type the fully qualified path to the executable file.

The status box in the **Run Program at User's Workstation** dialog box displays the current status of the program that is running on the client. For example, if the client runs the command successfully, the status reads **Executed**. If the command fails, the agent reports an error. To observe the results of running the executable file, you can establish a Remote Control session with the client.

Important

When an administrator uses Remote Execute to perform operations on the client, the user who is logged on to the client will also have elevated permissions and can then gain access to the same directories and files as the administrator. To maintain security, it is recommended that you use Remote Execute primarily to perform critical operations. You should also shut down any applications that you start during a Remote Execute session by initiating a Remote Control session. For more information, see the "Remotely Controlling Clients" section earlier in this chapter.

Transferring Files to and from Clients

If you discover a corrupt or missing file on a client, you can use File Transfer to transfer files directly to the client. You can also use File Transfer to transfer client files to your computer for troubleshooting purposes. To use File Transfer, establish a Remote Tools connection, and then click **File Transfer**.

When a directory tree appears for both the client and the administrator's computer, you can create new folders and copy, transfer, and delete files on the client directory.

Note

You should use File Transfer to move only small files, such as log files. You should not use it to move larger files or entire folders.

Restarting Remote Clients

When you replace a file or make configuration changes to a client, you might need to restart the client for those changes to take effect. There are two ways that you can remotely restart a client. You can establish a Remote Control session and then restart the client by using the **Shut down** command on the client's **Start** menu. Or, you can establish a Remote Tools connection to the client and then restart the client by using the **Reboot** button, especially if bandwidth is a concern.

When you restart the client during a Remote Control session by using the **Shut down** command on the client's **Start** menu, you immediately lose the client connection for clients running Windows 2000 or later. If there is a program running on the client that requires user input before shutdown, the client waits for user input. This can be a problem in unassisted Remote Control sessions, in which no user is present. You can avoid this problem by first ensuring that all programs are shut down or that other problems do not prevent the shutdown of the client during the restart process.

When you restart a client by using the **Reboot** button, you lose the client connection immediately for clients running Windows 2000 or later. If there is a program running on the client that requires user input before shutdown, the client shuts down without waiting for user input and any unsaved data is lost. You can avoid this problem by first ensuring that all programs are shut down before restarting the client.

Using SMS Remote Tools at a Client

Unless you specify in the site-wide settings that users cannot change their **Policy** or **Notification** tab settings for a client, they can open **Remote Control** in Control Panel and use the **Remote Control Properties** dialog box to change these settings.

Client Policy settings

On the **General** tab in the **Remote Control Properties** dialog box, a user can specify the level of remote access that is allowed.

If a user specifies **Full** or **None**, administrators can use all or none of the Remote Tools functions on the clients, respectively. If a user specifies **Limited** remote access, the administrator can use only the Remote Tools functions that the user specifies. For more information about Remote Tools functions, see the "SMS Remote Tools Overview" section earlier in this chapter.

Client access permission settings

On the **General** tab in the **Remote Control Properties** dialog box, the user can specify whether the Remote Control Agent displays a message each time that an administrator attempts to access the client to perform any remote function.

If the user selects **Pop up a window to ask for permission each time**, the Remote Control Agent displays a message that asks the user whether an administrator can remotely perform a specific task on the client. If the user grants permission by clicking **Yes,** the administrator is allowed access. If the user clicks **No,** the message closes and the administrator is denied access. If the user at the client does not respond to the message within 30 seconds, the administrator is automatically denied access.

If the user selects **Do not ask for permission**, an administrator is automatically permitted to access the client and perform any remote function. The user on the client is not notified unless the administrator initiates a Remote Control session.

Client Remote Control notification settings

On the **Notification** tab in the **Remote Control Properties** dialog box, the user can specify that the Remote Control Agent provide visual or audio notification whenever a Remote Control session is active on the client. The user can choose to:

- Display a visual indicator either as an icon in the notification area or as a high-security icon on the client desktop. The user can reposition the high-security icon on the desktop by dragging the icon or by right-clicking the icon to open a shortcut menu.
- Display the visual indicator only when a Remote Control session is active or at all times.
- Play a sound when the Remote Control session begins and ends or play repeatedly while the Remote Control session is active.

For more information about these options, see the "Notification Tab" section earlier in this chapter.

User control during a Remote Control session

During a Remote Control session, the user can open the **Remote Control Status** dialog box to view information by double-clicking the **Remote Control** notification icon in the notification area or on the client desktop. The user can also end the session by clicking **Close Session**. The **Remote Control Status** dialog box provides the following information:

- The version of the Remote Tools Client Agent that is running on the client
- The network protocol and address for the session
- The computer name of the client
- Whether video acceleration is enabled and the level of video compression
- The name of the administrator and the computer that established the Remote Control session

Note

Even after a Remote Control session has ended, a user can double-click the icon and view the name of the user and the computer that last established a Remote Control session with the client.

Advanced Features of SMS Remote Tools

The following sections describe some of the more advanced technical aspects of conducting Remote Control sessions:

- Role of Wuser32.exe on Clients
- Client Security Settings
- Client Hardware Settings
- Video Acceleration
- Improving the Performance of SMS Remote Tools

Role of Wuser32.exe on Clients

The Remote Control Agent, Wuser32.exe, is the key component for conducting all remote control operations and most other Remote Tools functions on clients. To determine whether the agent is started on clients running Windows NT 4.0 or later, you can use the **Processes** tab in Windows Task Manager. To determine whether the agent is started on clients running Windows 98, you can use the client's Control Panel. In Control Panel, double-click **Remote Control**, and then click **Show Status**. If the **Remote Control Status** dialog box opens, the agent is running. You can also use the client's Control Panel as an alternative way to determine whether the agent is started on clients running Windows NT 4.0 or later.

Wuser32.exe starts and runs in different ways, depending on the client's operating system.

On Clients Running Windows NT 4.0 or Later

On clients running Windows NT 4.0 or later, Wuser32.exe runs as a standard service. This service appears as **SMS Remote Control Agent** in the **Services** list. By default, its startup type is set to **Automatic**. Because Wuser32.exe is a standard Windows service, you can use the **net start** or **net stop** commands to stop and restart Wuser32.exe. When you use these two commands, either the full service name (SMS Remote Control Agent) or the short name (Wuser32.exe) works.

Note

You need administrative credentials to start or stop this service.

To stop and restart Wuser32.exe

1. To stop the service, type **net stop wuser32** at the command prompt.
2. To restart the service, type **net start wuser32** at the command prompt.

Note

If, for testing purposes, it is necessary to run the Remote Control Agent as a non-service (which places the agent in the context of the logged-on user) on a client running Windows NT 4.0 or later, use the following command option: wuser32 /nosvc.

On Clients Running Windows 98

On clients running Windows 98, Wuser32.exe runs as a background application, instead of a service. Because of this, Wuser32.exe does not appear in the process list in Windows Task Manager. Wuser32.exe runs as a child process that is started by SMS Client Services (Clisvc95.exe) under the **RunServices** registry key. This is why you do not find Wuser32.exe listed under the regular **Windows \Run** and **\RunServices** registry keys. You can stop and start Wuser32.exe manually by running the Wuser32.exe file from the command line.

To stop and restart Wuser32

1. To stop the service, change the directory to %*SystemRoot*%\MS\SMS\Clicomp\RemCtrl at the command prompt, type **wuser32 /x**, and then press ENTER.
2. To restart the service, change the directory to %*SystemRoot*%\MS\SMS\Clicomp\RemCtrl at the command prompt, type **wuser32**, and then press ENTER.

Client Security Settings

Security settings for all clients are configured on a site-wide basis. You configure the security-related settings for all clients in the site by using the **Remote Tools Client Agent Properties** dialog box on the SMS site server. These settings include:

- An option to prevent users from changing **Policy** and **Notification** tab settings on the clients.
- The **Permitted Viewers** list that defines who can remotely access clients in addition to members of the local administrators group.
- Visual and audio indicators to alert users when a Remote Control session is active.
- The requirement to request user permission before Remote Tools functions can be performed.
- The level of Remote Tools functionality that is allowed for clients in the site.

However, it is also possible to locally configure security settings for both the Legacy Client and the Advanced Client. The approach for managing the security settings for each type of client is discussed in the following sections.

Legacy Client Security Settings

The Security Munger manages all security-related settings for the Legacy Client. The Security Munger runs when the SMS site passes down new security settings to Legacy Clients. The Security Munger also overrides the local client settings with the site-wide settings if there are any differences. In SMS 2.0, the Security Munger reconciled security settings for clients assigned to multiple sites. In SMS 2003, Legacy Clients are allowed only a single site assignment.

Note

To run the Security Munger manually, at the command line on the client, enter **%*SystemRoot*%\MS\SMS\Clicomp\Remctrl\Rcclicfg.** If site-wide changes do not appear to take effect, reset the value in the **\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SMS\Client\Client Components\Remote Control\Combined Sites\<*site_code*>\LastChangedAt** key in the client registry to **0**. Then, run the Security Munger again. Using a **LastChangedAt** value of **0** causes a full security update.

Advanced Client Security Settings

The Remote Tools Client Agent manages all security-related settings for the Advanced Client. The CCM Policy Agent checks its management point and transfers the site-wide settings to the client by using the SMS WMI policy on the client. The functions of the Remote Tools Client Agent are similar to those of the Security Munger for Legacy Clients. However, because the Remote Tools Client Agent uses the SMS WMI policy, you have greater flexibility in managing the client configuration. With the Advanced Client, you can apply the SMS local policy by creating and compiling a Managed Object Format (MOF) file on the client. By using a MOF file to set the SMS local policy, you can choose whether to use the local policy or the site-wide policy for each Remote Tools setting. For more information, see the SMS 2003 Software Development Kit at http://www.microsoft.com/smserver/default.asp.

Disabling Site Settings

It is generally recommended that you leave the Security Munger enabled on Legacy Clients and the Remote Tools Client Agent enabled on Advanced Clients. Doing so ensures that any local changes to the registry are overwritten by the site-wide settings. However, in some situations you might want to keep local settings from being overwritten. For example, if you use the site-wide setting that requires user permission to perform Remote Tools functions, this can cause a problem for servers or other clients when a user is not present to respond to an administrator request.

To prevent the local settings on clients from being overwritten by the site-wide settings, you can create a value named **UpdateEnabled** in the client's registry under **\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SMS \Client\ Client Components\Remote Control** and set the value to **NO**.

Note

This value is not case-sensitive.

This option works for both Legacy and Advanced Clients. However, using the SMS local policy is recommended for this purpose, instead of modifying this registry key. The local policy gives the ability to selectively override individual settings on the client from those specified for the site.

Client Hardware Settings

The **Advanced** tab in the **Remote Tools Client Agent Properties** dialog box contains a number of hardware-related settings. You specify these settings for all clients in the site. These settings include:

- The default compression type for Remote Control sessions. The default setting is **Automatically Select**.

- The default remote access protocol.

> **Note**
> This setting is enabled only if you are configuring an SMS 2.0 site. SMS 2003 sites use only TCP/IP.

- Video acceleration for Windows-based clients, which is selected by default.
- The list of compatible video drivers for clients running Windows NT 4.0.

For most installations, the default settings in the **Remote Tools Client Agent Properties** dialog box should not be changed. If you experience problems during Remote Control sessions, such as the feature is not working, the client displays a blue or blank screen, or the client stops responding, the problems might be related to video acceleration or the type of video compression that you are using. For more information, see the "Video Acceleration" and "Video Compression" sections later in this chapter.

The following sections describe how the site-wide hardware settings are applied to the Advanced Client and the Legacy Client.

Advanced Client Hardware Settings

For the Advanced Client, the Remote Tools Client Agent manages all hardware-related settings. The Remote Tools Client Agent causes the site-wide settings that you specify to be used on the client. These settings are passed to the client when the CCM Policy Agent polls its management point. If you change the settings on the **Advanced** tab, those changes take effect for subsequently installed and previously installed Advanced Clients.

The Advanced Client always uses high (LZ) video compression. For more information, see the "Video Compression" section later in this chapter.

Legacy Client Hardware Settings

For the Legacy Client, the Hardware Munger manages all hardware-related settings. The Hardware Munger causes the site-wide settings that you specify to be used on the client. If the site-wide compression setting is **Automatically Select**, the Hardware Munger also determines the compression type for clients running Windows NT 4.0 or Windows 2000. If the site-wide setting is **Automatically Select**, clients running Windows NT 4.0 use low (RLE) compression and clients running Windows 2000 use high (LZ) compression.

Because hardware setting updates can change low-level functions, such as video acceleration, the Hardware Munger runs only when the Remote Tools Client Agent components are installed on the client and any time that you run **Repair Installation** from **Systems Management** in Control Panel. This is why the settings on the **Advanced** tab take effect only for subsequently installed Legacy Clients and not for previously installed Legacy Clients.

Changing advanced settings for previously installed clients

If you enable and configure the Remote Tools Client Agent for the site, and then later you want to change some of the settings on the **Advanced** tab in the **Remote Tools Client Agent Properties** dialog box, you have three options.

- You can disable Remote Tools for the entire site and wait until the next CCIM maintenance cycle (at least 25 hours) for the Remote Tools Client Agent components to be uninstalled from all clients. You can then change the **Advanced** tab settings as necessary, re-enable Remote Tools for the site, and wait until the next CCIM maintenance cycle for the Remote Tools Client Agent components to be reinstalled on all clients. This method, although easy, might not be suitable because of the loss of Remote Tools functionality.
- There are two client-side solutions for updating the hardware-related settings on previously installed clients. You can run the Hardware Munger manually from the client by using a command-line option. This makes the Hardware Munger function as though the client has just been installed. To do this, run the **Rchwcfg.exe install** command-line option from the %*SystemRoot*%\MS\SMS\Clicomp\Remctrl directory on the client.

  This executable file can also be run as an SMS software distribution package, which can be advertised to all clients that need to be updated.
- You can use the **Systems Management** icon on the client. You must first change the **Advanced** tab settings as necessary on the site server and wait until after the next CCIM maintenance cycle (at least 25 hours) for the settings to be moved down to all clients. Then, in Control Panel on the client, double-click the **Systems Management** icon, and then click **Repair Installation**. This reinstalls the Remote Tools Client Agent components, which updates the hardware-related settings on the client with the latest site-wide settings.

SMS 2.0 clients restricted to a single protocol

Although computers running the SMS Administrator console attempt to connect to SMS 2.0 clients by using all available protocols, SMS 2.0 clients can listen on only a single protocol. The client protocol for SMS 2.0 clients is a site-wide setting. If you specify a site-wide client protocol, and that protocol is not available on an SMS 2.0 client, there is no functionality on the client to use other available protocols. SMS 2003 clients use only TCP/IP.

Video Acceleration

For clients running Windows NT 4.0 or later, video acceleration reduces the work that is associated with each client screen refresh during a Remote Control session, which significantly speeds up the session.

On clients running Windows 2000 or later, video acceleration is not dependent on the type of video driver on the client. Video acceleration on clients running Windows 2000 or later can activate and run with any client video driver. On clients running Windows NT 4.0, video acceleration is dependent on the type of video driver on the client. This is key difference between video acceleration on clients running Windows 2000 or later and on clients running Windows NT 4.0.

To use video acceleration, you must enable this feature on the SMS site server.

► To enable video acceleration on the SMS site server

1. In the SMS Administrator console, navigate to **Client Agents**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
    └ Site Settings
     └ Client Agents
```

2. In the details pane, right-click **Remote Tools Client Agent**, and then click **Properties**.
3. On the **Advanced** tab, select **Install accelerated transfer on clients**.
4. Click **Apply,** and then click **OK**.

Video Compression

Video compression is an important aspect of video acceleration. Remote Tools uses video compression to reduce the size of screen-capture data that is being transmitted across the network during a Remote Control session. This minimizes the effect on network bandwidth. You can enable and configure the video compression properties on the **Advanced** tab in the **Remote Tools Client Agent Properties** dialog box. For more information, see the "Configuring Site-wide Settings" section earlier in this chapter.

There are three video compression options in SMS:

Low (RLE) Low, Run Length Encoding (RLE) compression compresses screen data, but not as effectively as high compression. You should use RLE compression for clients running Windows NT 4.0.

High (LZ) High, Lempel-Ziv (LZ) compression provides greater data compression than low compression, but it is primarily for clients with high-speed processors. Clients running Windows 2000 or later achieve better compression with LZ compression. LZ compression should not be used for clients with slow processors. LZ compression can be used only if video acceleration has been successfully loaded on the client, even if the client registry indicates that high compression should be used (compression = 1).

Automatically Select If you use the **Automatically Select** option, which is the default setting, SMS determines the best compression option to use based on the client type and CPU as follows:

- Advanced Clients always use high compression
- Legacy Clients running Windows 98 always use low compression

- Legacy Clients, which are Windows NT computers, use Pentium CPUs with at least 150 MHz as a threshold. Legacy Clients use low compression if they are below the threshold and high compression if above the threshold.

Note

Problems with Remote Control sessions, such as a blue screen or a blank screen, are often associated with LZ compression usage. If you experience such problems, try using RLE compression.

Video Acceleration on Clients Running Windows 2000 or Later

If video acceleration is enabled on a site-wide basis, all clients running Windows 2000 or later can be accelerated. The **Video Drivers** box on the **Advanced** tab in the **Remote Tools Client Agent Properties** dialog box is not relevant to video acceleration on clients running Windows 2000 or later.

For clients running Windows 2000 or later, there are four client component files involved in video acceleration:

- Idisw2km.sys — the SMS Mirror driver
- Idisw2km.inf — the file used to install the Mirror driver
- Wuser32.exe — the Remote Control Agent
- RCSvcs.exe — the Remote Control Services Manager

Installation of Video Accelerator Drivers for Clients Running Windows 2000 or Later

For clients running Windows 2000 or later, the Remote Control Services Manager performs the video acceleration driver installation. During the installation of the Remote Tools Client Agent, the Remote Control Services Manager:

1. Verifies that video acceleration is enabled site-wide.
2. Installs the SMS Mirror driver that is used for video acceleration.

Because Windows 2000 or later uses Plug and Play drivers, it is not necessary to restart the client after video acceleration is installed. The SMS Mirror driver is ready to use immediately after installation.

Note

If you uninstall the Remote Tools Client Agent, it is necessary to restart the client to remove the SMS Mirror driver. If you upgrade the driver, it is not necessary to restart the client.

You can verify the installation of the Mirror driver by viewing the Remote Control Services Manager section of the Remctrl.log file. The Remctrl.log file is located on the client in the *%SystemRoot%*\MS\SMS\Logs directory.

Note

When the Remote Control Services Manager installs the SMS Mirror driver, the client's screen will momentarily flash to a black screen and then return to normal.

Video Acceleration on Clients Running Windows NT 4.0

Video acceleration on clients running Windows NT 4.0 reduces the work that is associated with each screen refresh. Without video acceleration on clients running Windows NT 4.0, the entire screen is captured and sent each time a **DesktopChange** event occurs. The resulting bitmap is compressed and then passed across the network to the SMS Administrator console on the viewing computer. Video acceleration on clients running Windows NT 4.0 speeds the process by capturing only the rectangular region of the client's screen where changes have occurred. This reduces the size of each screen capture and increases the rate at which desktop changes can be passed across the network to the viewing computer.

For clients running Windows NT 4.0, there are four client component files involved in video acceleration:

- Idisntkm.dll — the accelerator driver that works together with the client's video driver
- RCHELP.sys — the accelerator helper driver that determines video driver compatibility
- Wuser32.exe — the Remote Control Agent
- Rchwcfg.exe — the Hardware Munger

The following factors determine whether video acceleration can be used on a client running Windows NT 4.0:

- You must enable video acceleration on a site-wide basis. You can do this on the **Advanced** tab in the **Remote Control Client Agent Properties** dialog box.
- The client's video driver must be included in the list of supported video drivers. For more information, see the “Video Drivers That Can Be Accelerated for Clients Running Windows NT 4.0” section.
- Windows NT 4.0 must determine that the IDISNTKM driver is compatible with the client's video driver.

Video Drivers That Can Be Accelerated for Clients Running Windows NT 4.0

The video drivers that have been tested and that are supported for clients running Windows NT 4.0 are listed on the **Advanced** tab in the **Remote Tools Client Agent Properties** dialog box on the SMS site server. You can add new drivers to this list, but you should test the results in a lab before implementing the change site-wide. Deleting items from this list makes them unavailable for video acceleration, even if video acceleration is enabled site-wide. For example, you might need to remove a specific driver if the manufacturer's video driver is incompatible with video acceleration for SMS Remote Control.

For clients running Windows NT 4.0, the list of supported video drivers is passed down to clients and added to the following registry key:

\HKEY_LOCAL_MACHINE\…\Sites\System\<*Site_code*>\Client Components\ Remote Control

The accelerator driver (Idisntkm.dll) controls video acceleration during a Remote Control session on clients running Windows NT 4.0. The driver is installed into the System32\drivers directory and then loaded and used concurrently with the video card manufacturer's video driver. Although the driver is loaded and running, it is used only during an accelerated Remote Control session.

After the Remote Tools Client Agent is installed on the client, Windows NT 4.0 determines whether a client's video card can be accelerated during the next restart.

To determine the client video driver

1. On the client, run Regedt32.
2. Navigate to the following registry key:

   \HKEY_LOCAL_MACHINE\Hardware\Devicemap\Video
3. Check each of the **\Device\Video0** keys and make note of the …**\Services\<*video driver*> \Device0** key. The **<video driver>** portion of the key is the video driver name as determined by Windows NT 4.0.

> **Note**
> You can ignore the **VGASave** entry. It is reserved for VGA Safe Mode.

To add the client video driver to the list of supported video drivers

1. In the SMS Administrator console, navigate to **Client Agents**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
   └ Site Settings
    └ Client Agents
```

2. In the details pane, right-click **Remote Tools Client Agent,** and then click **Properties**.
3. In the **Remote Tools Client Agent Properties** dialog box, click the **Advanced** tab.

4. Click the **New** button (gold star) to add a video driver name.
5. In **Video driver name** box, type the new video driver name, and then click **OK**.

Note

If acceleration is not available for a video driver that is used in your organization, experiment with the video driver on a single computer before adding an entry to the video drivers list for the entire site.

Adding unsupported video driver names to the supported video driver list can cause unexpected results if the video driver has not been tested for compatibility with video acceleration.

When you add a new video driver, the restrictions that are associated with changing the settings on the **Advanced** tab still apply. Only newly installed clients are affected by the changes to these settings. For more information, see the "Legacy Client Hardware Settings" section earlier in this chapter.

Determining Video Driver Compatibility for Clients Running Windows NT 4.0

During the installation of the Remote Control Agent components on a client running Windows NT 4.0, the Hardware Munger checks the client's video driver against the list of supported video drivers. This list is specified on the **Advanced** tab in the **Remote Tools Client Agent Properties** dialog box. If the Hardware Munger determines that there is a match, it inserts the accelerator driver into the registry to be implemented during the next restart.

During the restart, Windows NT 4.0 (not SMS) performs a test to determine if the client's video driver is compatible with the accelerator driver. If this test is successful, the accelerator driver and the client's video driver are loaded. If this test fails, Wuser32.exe removes IDISNTKM from the registry and client's video driver is not tried again. If Windows NT 4.0 loads the accelerator driver, but you still have display problems, try updating to the latest drivers from the video card manufacturer. This action resolves most video card driver problems.

Caution

Modifying the registry keys to prevent Windows NT 4.0 from determining its compatibility with IDISNTKM can cause unpredictable results.

The following steps explain the installation of the Windows NT 4.0 accelerator driver:

1. When the Remote Tools Client Agent is installed, the Hardware Munger adds all necessary IDISNTKM entries to the video driver registry key.
2. When the client is restarted, RCHELP.sys runs during startup. It reads the video driver registry key and creates a file in the %*SystemRoot*%\System32 directory called Viddrv.rch. You can view the contents of Viddrv.rch by using Notepad or another text editor.
3. Idisntkm.dll loads and examines Viddrv.rch to determine which driver to load. It uses the first driver in the registry list.
4. If Idisntkm.dll can load during the startup, it remains running as a video driver. No changes are made to any files or registry entries.

5. After the client completes the startup process and the Windows NT services start, Wuser32.exe determines if IDISNTKM is loaded. If it is successfully loaded, Wuser32.exe attaches to IDISNTKM and uses it to provide video acceleration. Otherwise, Wuser32.exe acknowledges that IDISNTKM is not loaded and removes the first IDISNTKM entry from the registry.
6. If acceleration successfully loaded during the last startup, it will continue to load without problems. If an IDISNTKM entry had to be removed from the registry during the previous startup, RCHELP.sys reads the registry again and then creates Viddrv.rch. Idisntkm.dll reads Viddrv.rch and attempts to load the next video driver in the list, if another entry is present, repeating steps 2 through 5 above.

The only scenario where acceleration might temporarily be lost is after a CCIM maintenance cycle, when the Hardware Munger is run again. This is primarily a problem for video cards with non-unified drivers. In this case, the registry is repopulated and the client must repeat steps 2 through 5 above until acceleration is successfully reloaded.

How Non-Unified Drivers Affect Video Acceleration for Clients Running Windows NT 4.0

There are two types of video drivers: unified drivers and non-unified drivers. Unified drivers require one set of drivers for all video modes. Non-unified video drivers require different drivers for each mode. Cirrus is one card manufacturer that does not use unified drivers and, therefore, requires drivers for each video mode.

The following examples show unified drivers and non-unified drivers in the **InstalledDisplayDrivers** key in the registry:

```
Cirrus:vga cirrus vga256 vga64k
Matrox:mga106
```

In this example, Cirrus lists separate drivers for each supported video mode, and Matrox lists only one driver for all supported video modes.

When the accelerator driver (IDISNTKM) is loaded, it is inserted into the registry between each driver entry. In the Cirrus example, IDISNTKM is inserted before each video mode. For the unified video drivers, such as Matrox, IDISNTKM is inserted only once. This results in the following updated registry keys:

```
Cirrus:idisntkm vga idisntkm cirrus idisntkm vga256 idisntkm vga64k
Matrox:idisntkm mga106
```

When the client restarts, Windows NT 4.0 tries the first driver in the **InstalledDisplayDrivers** key, together with IDISNTKM. If the two drivers work together, acceleration is enabled. With non-unified drivers, this process must be repeated as Windows NT 4.0 tries the driver for each of the supported video modes in succession. If acceleration fails for one of the drivers, Windows NT 4.0 discards that driver and the system then must be restarted to try the next driver. Although this might appear to be a problem with SMS Remote Tools, it is actually caused by the non-unified video driver architecture.

If you examine **HKLM\SYSTEM\CurrentControlSet\Services\Cirrus\Device0**, under the **InstalledDisplayDrivers** key, you can determine the state of attempted video acceleration for your card. For the Cirrus example, it might read as follows:

```
Vga idisntkm cirrus idisntkm vga256 idisntkm vga64k
```

This indicates that Windows NT 4.0 acceleration might be working with the non-unified Cirrus driver because an IDISNTKM entry is present in front of the Cirrus registry entry.

Usually, this process requires only one restart, because most video drivers are unified drivers. If you have clients that have older video cards with non-unified drivers, it might take multiple restarts to accomplish video acceleration.

To summarize:

- If you restart the client and Windows NT 4.0 enables acceleration, then IDISNTKM has been successfully loaded with the current driver.
- If a client has a video adapter that uses a non-unified driver, you might need to restart the client more than once to enable acceleration.
- If you have restarted the client multiple times and all drivers in the **InstalledDisplayDrivers** key have been attempted (including the final vga64k entry in the case of a non-unified driver), and acceleration still did not load, then acceleration cannot be used with this version of the manufacturer's video driver. Reinstalling the Remote Tools Client Agent components does not help in this situation, because it restarts the same process. Try updating to the latest drivers from the video card manufacturer.

Determining if Video Acceleration Is Installed for Clients Running Windows NT 4.0

After installing video acceleration on a client, you can confirm that the installation was successful by checking the registry.

► To determine if video acceleration is installed on a client running Windows NT 4.0

1. Using Regedt32 or Regedit, navigate to the following registry key:

   \HKEY_LOCAL_MACHINE\Hardware\Devicemap\Video

2. Review each of the **Device\Video*X*** keys (where ***X*** = the number of each display driver that is being used). Note the **Services\<*video driver*>\Device0** key for each display driver. The entry for **VGASave** should be ignored, because no attempt is made to accelerate the Safe Mode video driver.

3. Use these keys as pointers to view the following registry key:

   \HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services
   <key *value from the previous step*>\Device0

4. Check for the addition of IDISNTKM in the **InstalledDisplayDrivers** key to confirm that acceleration is loaded.

Alternatively, in Control Panel on the client, double-click **Remote Control**, and then click **Show Status**. The **Remote Control Status** dialog box opens and indicates whether acceleration is enabled.

Improving the Performance of SMS Remote Tools

There are a number of steps that you can take to enhance the performance of Remote Tools applications on your SMS site. These steps can reduce network bandwidth usage and increase the speed and efficiency of Remote Tools, particularly Remote Control sessions. The following sections describe several ways to enhance the performance of Remote Tools.

Remove Unnecessary Administrator Group Entries After Upgrading from SMS 2.0

SMS 2.0 and Windows NT 4.0 or later include multiple language-specific versions of the Administrator group. Because SMS 2.0 cannot determine which language-specific versions are required for a given SMS site, all localized versions of the Administrators group are added to the **Permitted Viewers** list. When you upgrade from SMS 2.0 to SMS 2003, these entries remain in the **Permitted Viewers** list.

To reduce network bandwidth usage and enhance the performance of Remote Tools, remove all unnecessary language-specific versions of the Administrator group from the **Permitted Viewers** list on the **Security** tab in the **Remote Tools Client Agent Properties** dialog box. Doing so enhances the performance of SMS Remote Tools by reducing the number of permitted viewers that are authenticated by the domain controller each time that you initiate a Remote Tools function.

Enable Video Acceleration

Enable video acceleration and, for clients running Windows NT 4.0, use the default set of tested and supported video drivers. Video acceleration works by sending an image of only the smallest rectangular area that includes all changes to the client's screen each time that it changes, instead of sending an image of the entire screen. Video acceleration works for clients running Windows NT 4.0 or later. For more information, see the “Video Acceleration” section earlier in this chapter.

Enable 16-Color Viewing

Enabling 16-color viewing significantly increases the speed of Remote Control sessions by reducing the color depth for clients that are using 256 colors or more. This feature is available on the **Control** menu of the Remote Control Client Viewer window.

To enable 16-color viewing, click the upper-left corner of the Remote Control Client Viewer window or press ALT+SPACEBAR to open the **Control** menu. Click **Configure**, and then select the **16 Color Viewing** check box in the **Control Parameters** dialog box.

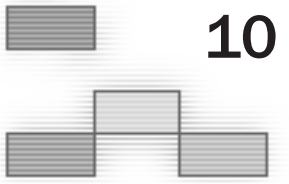
This option remains enabled for all Remote Control sessions until you disable it. While this feature is active, the client is visually unaffected, but the client desktop displayed within the Remote Control Client Viewer window uses only 16 colors.

Enable Wallpaper Suppression

You can also use the **Control** menu in the Remote Control Client Viewer window to select the **Suppress client wallpaper** check box. This feature causes clients to temporarily suspend their desktop wallpaper. When you complete the Remote Control session, the wallpaper is restored on the client. This feature is useful when you are conducting a Remote Control session with a client with high-color or elaborate background wallpaper. The **Suppress client wallpaper** option remains enabled for all Remote Control sessions until you disable it.

CHAPTER 10

Maintaining and Monitoring the Network



There are two situations in which network tools are indispensable: when you must diagnose network problems, and when you want to monitor and analyze patterns of network activity to avoid network problems. Microsoft® Systems Management Server (SMS) 2003 includes a set of useful network tools that help you monitor, capture, and interpret network data.

This chapter describes SMS network diagnostic tools, how they work, and how you can use them. Typically, you use Network Monitor to capture and analyze network frames to diagnose network problems and to identify optimization opportunities. You use Network Trace to graphically display site systems and the physical network that connects to them.

In This Chapter

- Using Network Monitor
- Using SMS Network Diagnostic Tools on Remote Computers
- Using Network Trace

Table 10.1 lists network monitoring and maintenance tasks and the SMS tools you use to accomplish those tasks.

Table 10.1 Network Monitoring and Maintenance Tasks and Tools

To do this task	Use this tool
Capture and examine network traffic (frames)	Network Monitor
Create capture and display filters to capture or view only the frames in which you are interested Automate data capture by using capture triggers	Network Monitor
Edit and retransmit frames onto your network	Network Monitor
Analyze and interpret captured data	Network Monitor Experts
Graphically map the network connections between site systems and network devices such as routers	Network Trace

Using Network Monitor

By using Network Monitor, you can capture frames directly from the network traffic data stream and examine them. You can use this information to analyze ongoing patterns of usage and diagnose specific network problems. To capture network frames, Network Monitor places the network adapter of the computer you are using into promiscuous mode. In promiscuous mode, all frames detected by the network adapter are transferred to a temporary capture file, regardless of the destination address of each frame.

Frames, also known as packets, are packages of information that are transmitted as a single unit over a network. Every frame follows the same basic structure and contains:

- Control information such as synchronizing characters.
- Source and destination addresses.
- Protocol information.
- An error-checking value.
- A variable amount of data.

You either can capture all the frames that pass by the network adapter or design a capture filter to capture only specific frames, such as those originating from a specific source address or using a particular protocol. When you begin capturing network data, the captured frames are stored in a temporary capture file. After the data capture process concludes, you can view the frames immediately or save the frames in the temporary capture file to a capture file. These files provide important diagnostic information to administrators and third-party support services. By default, the capture file name extension is .cap.

The default size of the temporary capture file is 1 MB. When the temporary capture file fills to capacity, the oldest frames captured are lost. If your temporary capture file fills too quickly and you begin to overwrite buffered data, increase the size of the temporary capture file. When you increase the size of the temporary capture file, you should consider the amount of RAM on your system. If the temporary capture file size exceeds the amount of RAM, some frames might not be captured while your system swaps memory to disk. You can use capture triggers to automatically stop the data capture process when the temporary capture file fills to a predetermined level. You can also reduce the amount of data placed in the temporary capture file during data capture by using capture filters.

Network Monitor captures only the traffic that passes through the network adapter of the computer it is running on. This means that you can capture only the traffic of the local network segment. If your network consists of different segments, you can use Network Monitor to connect to a computer on another segment that has the Network Monitor Driver installed. The Network Monitor Driver can be enabled in the protocols properties of a connection to capture the segment's traffic. For more information about using Network Monitor to capture traffic on a remote computer, see the "Using SMS Network Diagnostic Tools on Remote Computers" section later in this chapter.

Capture Triggers

You can use Network Monitor to configure capture triggers. During the capture process, a capture trigger monitors the network traffic data for one or both of the following trigger events:

- The temporary capture file fills to a specified level.
- A specific data pattern occurs in a captured frame.

When a trigger event occurs, the capture trigger can be configured to:

- Either sound an audible signal or stop capturing data.
- Run a program or a batch file.

For example, you might configure a trigger to stop capturing data when a specified hexadecimal or ASCII pattern is found in a frame, which preserves the captured frame in the temporary capture file.

Capture Filters

You can limit the frames that are captured by designing a capture filter. A capture filter compares the network traffic to a defined set of criteria, and copies frames that meet the criteria to the temporary capture file. You build a complete capture filter expression by specifying the protocols, address pairs, and data patterns of the frames that you want to include in, or exclude from, the capture.

Experts

Although you can examine captured frames to analyze network problems, complete and accurate analysis is difficult if you do not have a detailed knowledge of what your network traffic looks like. This knowledge requires examining data on a frame-by-frame basis, and knowing which network service generated each frame. Network Monitor includes a set of Experts, which are automated tools designed to help you interpret the information subtleties of captured network data.

Frames consist of a complex mix of addressing information, protocol information, and the actual data being transmitted across your network. This information is arranged in different layers. Each layer contains potentially useful information. For example, one layer contains the frame' destination address. By examining a frame's destination address, you can determine whether the frame was broadcast to all recipients on your network or sent to a single station. By examining each part of a frame, and by reviewing sequences of frames, you can determine exactly why each frame was generated.

For example, in a Microsoft Windows® 2000 network, when a computer is configured as a WINS client, it seeks a logon server by querying the WINS server for the domain name. The WINS server responds by sending a frame that contains the IP address of all registered domain controllers in its WINS database. The client then sends a directed frame to each server listed in the response, asking it to validate the logon request. Each server then sends a response frame to the client. The client then takes the first server response and initiates a series of frame sequences with the server to actually validate the logon.

This complex series of events illustrates why a knowledge of the various network services and the tasks they perform is essential to understanding what you see in each frame. The Network Monitor Experts assist you in performing sophisticated post-capture analysis of your network traffic. For more information, see the "Using Network Monitor Experts" section later in this chapter.

Before you run Network Monitor, ensure that the computer running Network Monitor meets the following requirements:

- A Windows 2000 Server or later operating system version is installed.
- Administrator rights have been granted to the Microsoft Windows 2000, Windows XP, or Windows Server™ 2003-supported user.
- Network Monitor is installed.

To install Network Monitor:

1. Insert the SMS 2003 product CD.
2. Click **Start**, click **My Computer**, right-click the product icon, and then click **Explore**.
3. Double-click the **Network Monitor** folder, double-click the **I386** folder, and then double-click Netmonsetup.exe.

- The computer includes a network adapter that supports promiscuous mode.

Using Network Monitor involves these tasks:

- Capturing network traffic
- Examining captured data
- Using Experts to analyze the captured data

Capturing Network Traffic

By using Network Monitor, you can capture all the network traffic that passes by your network adapter on the local subnet, or filter the traffic to analyze only the frames you are interested in.

There are several circumstances that might prevent Network Monitor from launching or compromise its performance.

One scenario is when Authenticated Users is manually removed from the Users group. To resolve this issue, add the specific Network Monitor user to the group. To complete the workaround, the user needs to log off and log back on to the computer.

Another scenario is when the Discretionary Access Control List (DACL) of the system directory is changed to disallow normal user's access. To resolve this issue, add the specific Network Monitor user to the DACL of the system directory. To complete the workaround, the user must log off and log back on to the computer.

Network Monitor runs with reduced access in which administrative privileges have been removed. If you receive an Access Denied message when you follow this procedure, add your user name to the permissions list of the file or folder that you want to access.

To start Network Monitor

1. On the **Start** menu, point to **All Programs**, point to **Microsoft Network Monitor**, and then click **Network Monitor**.
2. To begin capturing data, on the **Capture** menu, click **Start**.

   When the Frame Viewer window opens, you can view a summary listing of captured frames.
3. To stop the data capture, on the **Capture** menu, click **Stop and View**.

The network traffic you capture is the traffic passing by your computer on your local subnet. Frames that run on another subnet are typically never routed to your subnet unless they are broadcast or the destination address is a computer on your subnet.

Note

It is not recommended to capture local network data from your site server. Placing your network adapter into promiscuous mode is a processor-intensive process and can adversely affect the performance of other processes on the server. If you want to run Network Monitor on the site server as a client for remote capture of network data, it will not cause a performance issue. For more information, see the "Using SMS Network Diagnostic Tools on Remote Computers" section later in this chapter.

Examining Captured Data

In the Network Monitor Frame Viewer window, you can view individual frames in detail by double-clicking any frame. By examining the constituent parts of a frame, you can learn, for example, whether the frame was broadcast or directed and which properties are associated with each part of the frame. You can also discover which protocols the frame was using and where the frame originated and why it was sent.

When you double-click a frame, the Frame Viewer window splits into three panes. The top pane is the Summary pane, which displays general information about the captured frames in the order that they were captured. The frame that you have selected to examine is highlighted in the Summary pane. To examine another frame, scroll to it and then click it. The middle pane is the Detail pane, which parses the network frame data and displays the individual layers in more detail. You can expand or collapse the details of each layer by clicking the plus (+) and minus (-) symbols in the Detail frame. The bottom pane is the Hex pane, which displays the frame data in hexadecimal and ASCII format.

Using Network Monitor Experts

You can run the Network Monitor Experts supplied with Network Monitor, third-party Experts, or custom Experts that you create yourself. For more information about creating Experts, see the Platform SDK at http://support.microsoft.com/support/smsmgmt/content/sms20sdk.asp.

Table 10.2 lists the functionality of the Experts supplied with Network Monitor.

Table 10.2 Network Monitor Experts

To perform this task	Use this Expert
Calculate the average server response time for servers on a network subnet	Average Server Response Time Expert
Calculate frame statistics for a specified property found in frames in a capture file	Property Distribution Expert
Calculate statistics about the distribution of protocols found in frames in a capture file	Protocol Distribution Expert
Find all TCP frames that have been retransmitted to the same computer in a capture file	TCP Retransmit Expert
Determine the top senders and recipients in a capture file based on the source and destination addresses of each frame	Top Users Expert
Recombine data for a transaction that was sent across the network in multiple frames	Protocol Coalesce Expert

Example: Measuring network response time A common user complaint is that a network server or the network is slow. Slow response time problems are often frustrating to solve because it can be difficult to link server performance data to the server responsiveness that users experience at their desktop. Also, it is often difficult to obtain the information you need to determine whether network response times warrant changing configurations or adding additional servers.

Quantifying the speed of the network is simplified by using the Average Server Response Time Expert. This Expert uses Server Message Block, any specified TCP ports (such as HTTP), and any specified IPX sockets to calculate the number of seconds it takes for a server to respond to a client's request for data. You can run the Expert to establish a baseline of average server response times and then compare current responsiveness to historical data. This Expert is also a useful way to quantify server responsiveness under different configurations, such as when an existing Microsoft SQL Server™ computer is also configured as a WINS server.

► To measure average server response time

1. Start the Network Monitor Capture window.
2. To begin capturing frames, on the **Capture** menu, click **Start**.
3. To end the capture and view the summary list of captured frames, on the **Capture** menu, click **Stop and View**.
4. To open the Network Monitor Experts window, on the **Tools** menu, click **Experts**, and then click **Average Server Response Time Expert**.

5. To configure the Expert, click **Configure Expert** and specify the TCP ports and IPX sockets that the Expert should monitor.
6. Click **OK**, **Add to Run List**, and then **Run Experts**.

   The average response times of servers measured in the captured data appears in the Event Viewer window.

Using SMS Network Diagnostic Tools on Remote Computers

Network Monitor captures only the traffic that passes through the network adapter of the computer it is running on. This means that you can capture only the traffic of the local network subnet. You can gather statistics about network traffic on other subnets by installing Network Monitor Driver on a computer running a Windows 2000 or later operating system in another subnet, and then connecting to that computer remotely.

By connecting to a remote computer, you can initiate network traffic capture on the remote computer, and then view and save the data on your local computer. You can also configure and run capture triggers on the remote computer.

Before you use Network Monitor's remote capabilities on a remote computer, ensure that your system meets the following requirements:

Network adapter

The network adapter in the remote computer must support promiscuous mode.

Installation

Your local computer must run a Windows 2000 or later operating system. Make sure to install and enable the Network Monitor Driver on the remote computer. To install the driver, perform the following steps:

1. In Control Panel, double-click **Network Connections**, right-click the network connection, and then click **Properties**.
2. On the **General** tab, click **Install**.
3. In the Select Network Component Type window, click **Protocol**, and then click **Add**. Click **Network Monitor Driver**, and then click **OK**.

The Network Monitor Driver should now be installed and enabled. When you add a network protocol, the Network Monitor Driver service appears in the protocol listing.

Protocols

A connection-oriented protocol, such as TCP or NetBIOS, must be available on both the local computer and the remote computer.

Network Monitor installation

Network Monitor must be installed and running on your local computer. On the remote computer, you need only ensure that the Network Monitor Driver is installed on that computer.

Capturing Traffic on Remote Computers

Network Monitor captures only the traffic that passes through the network adapter of the computer that it is running on. This means that you can capture only the traffic of the local network subnet. You can gather statistics about network traffic on other subnets by installing the Network Monitor Driver on a computer running a Windows 2000 or later operating system in another subnet, and then connecting to that computer remotely. The remote computer performs all capture operations, transfers statistics to your local computer, and saves capture files to its own hard disk.

When Network Monitor connects to Network Monitor Driver on a remote computer, you can to use that computer's network adapter as though it were installed locally. When you use Network Monitor on the local computer, data capture, filters, and triggers function on the remote system just as they would locally. If you stop a remote capture and display the data, the capture data is displayed as if the capture were local. You can save the capture file to any location.

When Network Monitor connects to a remote computer running the enabled Network Monitor Driver and uses the computer to capture remote subnet traffic, the remote computer gives no visual indication that it is being used to capture traffic; it simply creates a capture file, which you can view on your local computer.

► To capture traffic on a remote computer

1. Connect to the remote computer that has the Network Monitor Driver enabled.
2. Start Network Monitor on the local computer.
3. On the **Capture** menu, click **Networks**, and in the **Networks** dialog box, expand the **Remote** node.
4. Double-click the **Double click for remote NPPs** line.
5. In the **Remote NPP Connection** dialog box, type the remote computer name or IP address and click **OK**.
6. If the remote computer has more than one network adapter installed, select a network adapter, and then click **OK**.
7. To begin capturing data, on the **Capture** menu, click **Start**.

   The capture window title bar displays the network adapter and computer name of the computer from which you are capturing data.

Using Network Trace

You can use Network Trace to create a network diagram for any SMS site system that you select. The network diagram that you create displays network connectivity from the perspective of the site system that you have selected, not from the perspective of the computer from which you are running Network Trace. You can use Network Trace to display the IP network connections of a remote site system. Also, you can use Network Trace to display the site system roles performed by the selected site system and by all the servers connected to that site system.

You can create network diagrams that display the following information:

- All servers connected to the selected site system
- Site system roles performed by each server
- Network devices such as routers
- IP subnets
- IP addresses

A network diagram displays information in either a trace view or a site view. In a trace view, only the site systems within the site database are displayed. In a site view, all known subnets and routers are also displayed, along with the site systems within the site database.

Network Trace creates network diagrams that are based upon information in the SMS site database. SMS gathers this information during the server and network discovery processes. SMS Server Discovery runs immediately after SMS installation and periodically thereafter to discover servers that you have configured as site systems. After Server Discovery runs, Network Trace can diagram the communication links between other servers and the site system you select. Network Discovery is not enabled by default. Also, you must schedule and configure Network Discovery to discover devices such as routers.

Note

To diagram devices outside your local subnet, you must run Network Discovery on all subnets in the site that you want to diagram. If you do not do this, network diagrams created by using Network Trace display only the local subnet.

To create a network diagram for a site system

1. In the SMS Administrator console, navigate to **Site Systems**.

```
Systems Management Server
 └ Site Database <site code - site name>
  └ Site Hierarchy
   └ <site code - site name>
   └ Site Settings
    └ Site Systems
```

2. Right-click a site system server, point to **All Tasks**, and then click **Start Network Trace**.

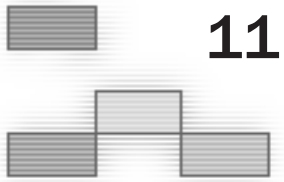
   The **Network Trace** window opens and displays a diagram of the IP communication links between the site system you selected and other servers and network devices that are connected to the selected site system.

Other features of Network Trace include the ping provider and the Component Poller. You can use the ping provider to transmit an Internet Control Message Protocol echo, which is more commonly known as a ping. You can send a ping to all devices displayed in the network diagram, or to only the devices that you select, to confirm the IP communication link. Pings are sent from the site server, not from the computer on which you are logged on. For the ping provider to function correctly, you must be able to connect to the site server. For a primary site, this means that you must have DCOM/WMI connectivity enabled on the site server. For a secondary site, you must be an administrator on the site system.

You can use the Component Poller to query the status of SMS components installed on the selected site server. You can use it to determine if a component is running, paused, or stopped, the last time the component was polled, and the component type. Like the ping provider, the Component Poller runs on the site server. For the Component Poller to function correctly, you must have the appropriate connectivity and SMS security rights to the site server. For a primary site, this means that you must have DCOM/WMI connectivity enabled on the site server and you also must have **Administer** permission for the **Site** object, which you set by using the **Security Rights** console item in the SMS Administrator console. For a secondary site, you must be an administrator on the site system.

CHAPTER 11

Creating Reports



Microsoft® Systems Management Server (SMS) 2003 generates a tremendous amount of network, inventory, discovery, and status information, which it maintains in your SMS site database. Depending on the level of each site in your SMS hierarchy, your site database might also include information that is passed up from child sites. One challenge that you face as an administrator is retrieving the pertinent data that is necessary to monitor and evaluate your SMS system and to help you and others effectively manage your organization. You can use SMS reporting to gather, organize, and present information that is collected in your site database.

SMS 2003 provides a number of predefined reports that you can use to gather important information from your site database. Administrators can create, manage, and secure reports by using the SMS Administrator console. Administrators and other report users, such as help desk specialists or business decision-makers, can run reports by using Report Viewer. Report Viewer is a browser-based application that runs with Microsoft Internet Explorer. You can create and administer reports in the secure environment of the SMS Administrator console and end users can run reports without the need to access an SMS Administrator console.

You can export and import reports by using the Export Object Wizard and Import Object Wizard. SMS 2003 exports reports by writing report object definitions, which are the properties that define a report, to a file. Only the report object definitions are exported or imported, not report data. You can use exported report files to share reports with other SMS administrators, or to import reports that you obtained from other SMS administrators or other sources.

You can also create dashboards, which are sets of reports in a grid that you can display in a single window of Report Viewer. You can use dashboards to monitor information about a variety of SMS objects or systems. You cannot export or import dashboards.

In This Chapter

- Understanding Reporting
- Working with Reports
- Working with Dashboards

Understanding Reporting

Reporting in SMS 2003 is integrated into the SMS Administrator console. Reports are secured SMS objects that you can create and manage by using the SMS Administrator console. Many predefined reports are provided with SMS 2003, and you can create additional reports by using the SMS Administrator console. You can also use the Import Object Wizard to import reports that are created outside of your SMS Administrator console. You can run reports by using Report Viewer, which is a browser-based application that you can start either from within the SMS Administrator console or by using a URL with Internet Explorer. Report users do not need to have access to an SMS Administrator console to view reports. The code for Report Viewer is located on a reporting point, which is an SMS site system role.

Note

You must enable a reporting point to use Report Viewer. For more information, see Chapter 15, "Deploying and Configuring SMS Sites," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*

Like other SMS objects, you must have the appropriate credentials to create, modify, delete, view, or run reports. For more information about report security, see Chapter 5, "Understanding SMS Security" in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

SMS 2003 provides a number of predefined reports that you can use to gather important information from your site database. For many administrators, these reports provide sufficient information to administer their computer infrastructure and SMS system. However, you might find that your information needs extend beyond the predefined reports. In this case, you can create your own reports or copy and modify predefined reports to better meet your needs.

The principal element of a report is a Structured Query Language (SQL) statement that defines which data the report gathers and returns as the result set. A result set is a tabular arrangement of the data in columns and rows. A report can also return multiple result sets. The SQL statement in a report does not run directly against your SMS site database tables. Instead, the SQL statement runs against a set of Microsoft SQL Server™ views, which point to records in your SMS site database tables. Each time that you run a report, the information returned consists of data that is current in the database at the time that you run the report. To create new reports by using the SMS Administrator console you must have a working knowledge of SQL. However, no knowledge of SQL is required to import new reports.

You can export reports from your SMS site database by exporting the report object definitions to Managed Object Format (MOF) files. You can also import MOF files that contain report object definitions into your SMS site database. This allows you to share your reports with other users and sites and to use reports that are created by others.

Reports are not propagated up or down the SMS hierarchy; they run only against the site's database of the site on which they are created. However, because primary sites contain inventory data from child sites, when a report retrieves data from a primary site's database, it might retrieve data that was forwarded from a child site.

Report Types

There are four types of reports:

Predefined reports A variety of reports are provided with SMS 2003 to help you quickly obtain information that is useful to the administration of your SMS operations, such as reports that provide information about the hardware inventory data in your SMS site database.

Predefined reports include, but are not limited to, reports in the following categories:

- Hardware
- Software
- Software distribution
- Software metering
- Software updates
- Network
- Operating system
- SMS site
- Status messages

Custom reports Reports that you create either by copying and modifying predefined reports or by creating new reports. To create a new report, you must specify an SQL statement that determines which records are returned when the report is run. For more information, see the "Creating and Modifying SQL Statements" section later in this chapter.

Supplemental reports Reports created outside of SMS 2003, which you can place in a designated folder on a reporting point to extend your reporting capabilities. These reports will primarily be Active Server Pages (ASP) pages. However, it can be any file that you can display by using Internet Explorer 5.0 or later. Because supplemental reports are not secured SMS objects, any user can view them unless you secure them by using Microsoft Internet Information Server (IIS) security.

Dashboards Sets of reports that are displayed in a grid within a single window of Report Viewer. You can use dashboards to quickly obtain information about a variety of topics.

Report Prompts

A *prompt* is a report property that you can configure when you create or modify a report. When a user runs the report, a prompt requests the user to enter a value for a required parameter prior to running the report. A report can contain more than one prompt.

You can use prompts to limit or target the data that a report retrieves. For example, you create a report that retrieves hardware inventory data for a given computer and prompts the user for a computer name. Report Viewer then passes the user-specified value to a variable that is defined in the SQL statement for the report. Provided that you have properly configured the SQL statement, the report returns hardware inventory data only for the specified computer. For more information, see the "Integrating Report Prompts" section later in this chapter.

To help report users enter prompt values, you can specify a default value for a prompt. You can also configure a prompt to display a list of appropriate values from which the user can choose. For more information, see the "Creating Report Prompts" section later in this chapter.

Report Links

You can use a link in a source report to provide users with ready access to additional data, such as more detailed information about each of the items in the source report. For example, you might link a report that lists all site codes to another report that lists all recent error messages for a given site code. The source report passes a specific site code to the target report based on which line item in the source report that the user chooses to obtain more information. A report can only be configured with one link, and that link can only connect to a single target.

Note

To take advantage of a report link, the user of the source report must also have the appropriate permissions to the link target. For example, if a report links to the Status Message Details page, a user must have **Read** permission for the Status Message object to view status message details. Or, if a report links to another report, the user must have instance-level **Read** permission for that report or class-level **Read** permission for the Report class to view the target report. For more information, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

You can link a source report to any of the following targets:

Another report This target can be any predefined or custom report. Links to supplemental reports are described later in this list. If the target report requires one or more prompts to run, the source report must contain a column with valid values for each prompt. You must specify the column number to use for each prompt. For example, you might link a report that lists computers that were discovered recently to a report that lists the last messages that were received for a specific computer. When you create the link, you might specify that column 2 in the source report contains computer names, which is a required prompt for the target report. When you run the source report, link icons appear to the left of each row of data. When you click an icon for a row, Report Viewer passes the value in the specified column for that row as the prompt value that is needed to display the target report. For more information, see the "Report Prompts" section earlier in this chapter.

Computer Details page This link is to the Computer Details page, which is a specialized page of Report Viewer. Many of the predefined reports provided with SMS 2003 are designated to appear on this page and are configured to display detailed information about a specific computer. However, you can designate any report that has one prompt or no prompts to appear on the Computer Details page. A source report that you link to the Computer Details page can contain a column with values that can be passed as the prompt parameter for reports that appear on this page. When you create the link, you specify the number of that column. When you run the source report, link icons appear to the left of each row of data. When you click an icon, Report Viewer opens the Computer Details page and automatically enters the value from the specified column of the row as a parameter for reports. You can then use this value to run reports on this page or you can enter another value. For more information, see the "Using the Computer Details Page" section later in this chapter.

Status Message Details page This link is to the Status Message Details page, which is a specialized page of Report Viewer. This page can only be accessed from a report that contains status messages. You can use the Status Message Details page to display information about a specific status message, based on the RecordID property for the message. The source report that you link to the Status Message Details page must contain a column with RecordID values. When you create the link, you specify the number of that column. When you run the source report, link icons appear to the left of each row of data. When you click an icon, the Status Message Details page opens and displays information about the specific message. For more information, see the "Using the Status Message Details Page" section later in this chapter.

Uniform Resource Locator You can use this target to link a source report to a supplemental report or to any file that is supported by HTTP. To create the link, you specify the URL of the target, which can be either an absolute or a relative URL.

You can also configure a URL link to pass column information from the source report as a parameter to the target report. To do this, you specify column values by using the syntax <*column_number*> in the URL, as in the following example:

```
CustomReport.asp?MachineName=<3>&Network=<5>
```

In the URL example, <3> is replaced with the value from column 3 and <5> is replaced with the value from column 5 in the source report. You must configure the target page to accept the data that Report Viewer passes to it. Report Viewer performs no syntax checking.

The URL that is specified in the report properties can be a maximum of 1,024 characters. When a report user clicks the link, and the source report data is inserted into the URL, the target URL can be up to 2,048 characters.

Changing linked reports

When you configure links, you create dependencies between the source report and its target. This is especially true when you create a link and specify the source report column that contains data the target needs to run. This is the case when the target is a report that has prompts or links to the Computer Details page or the Status Message Details page.

When you create such a link, and then delete or change the order of columns in the source report, you can break the link. For example, suppose that you link a source report to the Status Message Details page. In the link, you specify column 2 of the source report as the column that contains RecordID, which is the value that the target needs to run. Subsequently, you change the SQL statement for the source report so that RecordID values are returned in column 3 and site codes values in column 2. If you run the source report again, Report Viewer passes the data in column 2, which is now the site code data. Because the Status Message Details page needs a RecordID to run, it returns no data. To prevent this, any time that you change the order of columns in a source report, you should also change the link properties to reflect the changes made to the columns.

You can also break links by adding, deleting, or changing a prompt in a target report. Because one or more source reports can pass data that is required by a prompt or prompts in a target report, such changes can break several links. To prevent this, when you change prompts in a target report, you need to change the link properties to reflect the prompt changes in any reports that link to the target report.

Hyperlinks based on report value data

In addition to the links described earlier, which you can configure when creating a report, hyperlinks can also appear in a report when it is run. These hyperlinks appear only when report values of a specific format are returned in the result set of the report query.

For report values that begin with **http:**//, **ftp:**//, **file:**//, or \\, Report Viewer converts the entire text string into a hyperlink. This can provide you with an additional way to redirect report users to additional information.

Note

Only report values that begin with the prefixes **http://**, **ftp://**, **file://**, or **** are converted into hyperlinks. There is no support for embedded URLs within text, multi-URLs, or a mixture of URLs and text.

Working with Reports

SMS 2003 provides you with a number of predefined reports that you can use to quickly gather a wide variety of information about your SMS operations. You create and manage reports by using the SMS Administrator console. You run and display the results of a report by using Report Viewer, which requires Internet Explorer 5.0 or later. You can view and navigate the list of reports by using either the SMS Administrator console or Report Viewer.

This section includes information about:

- Creating and managing reports.
- Creating and modifying SQL statements.

Before you can begin using SMS reporting, you must enable one or more of your site systems as a reporting point. A reporting point is a site system that hosts the code for Report Viewer and any supplemental reports.

SMS 2003 does not automatically enable reporting points. You must enable all reporting points as required to provide access to reports in your site. To balance a heavy demand for reports in a larger site, you can enable more than one reporting point and then point different groups of users to different URLs for each reporting point. When you start Report Viewer from the SMS Administrator console, you select the specific reporting point that you want to use.

For more information about how to create an SMS site system and enable a reporting point, see Chapter 15, "Deploying and Configuring SMS Sites," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Creating and Managing Reports

You must have **Create** permission for the Reports security object class to create or import reports. You must also have the appropriate permissions for the Reports security object class or instance to modify, delete, export, or run a report. For more information about permissions, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

The tools that you can use to complete the various tasks of creating and managing reports are described in Table 11.1.

Table 11.1 Tools for Creating and Managing Reports

Tool	Task
SMS Administrator console	Create, modify, delete, export, or import reports
SMS Administrator console or Report Viewer	View the list of available reports
Report Viewer (can be launched from the SMS Administrator console)	Run reports
Report Viewer	Run reports on the Computer Details page
Report Viewer	View and run supplemental reports
Report Viewer (Report Results page)	Print a result set, save it as a comma-delimited file, or copy it to the Clipboard
Report Viewer (Report Results page)	Bookmark a report as a favorite or send a link to a report in an e-mail

Viewing the List of Reports

You can view the list of available reports by using either the SMS Administrator console or Report Viewer.

To view the list of reports by using the SMS Administrator console

- In the SMS Administrator console, navigate to **Reports**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Reports
```

The list of reports for which you have **Read** permission appears in the details pane.

In the SMS Administrator console, you can sort reports by **Name**, **Category**, or **Report ID**. To sort the list of reports, click the appropriate column heading. You can also filter which report categories appear and choose or change the order of the columns in the details pane of the SMS Administrator console. These filters apply only to the local computer on which the SMS Administrator console is running.

To filter the list of reports by using the SMS Administrator console

1. In the SMS Administrator console, navigate to **Reports**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Reports
```

2. Right-click **Reports**, point to **All Tasks**, and then click **Filter Reports**.
3. In the **Filter Reports** dialog box, select one or more categories in the **Categories** list, and then click **Display/Hide**.

   In the **Categories** list, the **Display** column value for the selected category or categories switches between **Yes** (Display) and **No** (Hide).

To view the list of reports by using Report Viewer

1. In the SMS Administrator console, navigate to **Reports**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Reports
```

2. Right-click **Reports**, point to **All Tasks**, and then point to **Run**.
3. On the **Run** menu, click the name of the reporting point that you want to use to start Report Viewer.

   Report Viewer starts on the main page.

Note

You can also start Report Viewer on its main page by typing the designated URL for a reporting point in the **Address** box of Internet Explorer. If you have the appropriate credentials, you can see the URL for a reporting point on the **Reporting Point** tab in the **Site System Properties** dialog box.

4. On the Report Viewer main page, perform one of the following procedures:
   - In the reports tree, expand a category to view a list of reports in that category for which you have Read permission. To run a report, click the report, enter values for any required parameters, and then click Display.
   - To view the list of dashboards, click Dashboards. For more information, see the "Running Dashboards" section later in this chapter.
   - To view the list of reports that are designated to appear on the Computer Details page, click Computer Details. Only reports for which you have Read permission appear on this page. For more information, see the "Using the Computer Details Page" section later in the chapter.
   - To view the list of supplemental reports, expand Supplemental Reports. For more information, see the "Using Supplemental Reports" section later in this chapter.

> **Note**
> The **Supplemental Reports** item appears only if you place at least one supplemental report in the designated folder on the reporting point.

Running Reports

You run reports by using Report Viewer. You can start Report Viewer from the SMS Administrator console by right-clicking a report, choosing a reporting point, and then clicking **Run**. You can also start Report Viewer by entering a report's unique URL in the **Address** box of Internet Explorer or by entering the URL of the Report Viewer main page on a reporting point in the **Address** box of Internet Explorer.

You can also use a report's URL to schedule the report to run automatically at a specified time. For more information, see the "Scheduling Reports" section later in this chapter. This can be helpful for reports that can take a long time to run, such as a report that returns a large amount of data. You can schedule such reports to run at a time when your network is less busy.

The following procedure describes how to run individual reports starting from the SMS Administrator console. For information about running reports by using the **Computer Details** page of Report Viewer, see the "Using the Computer Details Page" section later in this chapter. For information about running supplemental reports, see the "Using Supplemental Reports" section later in this chapter.

To run a report from the SMS Administrator console

1. In the SMS Administrator console, navigate to **Reports**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Reports
```

2. Right-click the report that you want to run, point to **All Tasks**, and then point to **Run**.

3. On the **Run** menu, click the name of the reporting point that you want to use to start Report Viewer.

   If the report has prompts, Report Viewer starts at the Report Information page for the selected report, where you can enter values for any required parameters, and then click **Display**. Click **Values** to display a list of values that can be entered in the prompt. You can also click **Show tree** on the menu bar to display the full list of reports.

Important

The number of values that might be returned when you click **Values** can be very large and is limited by default to 1,000. For information about how you can change the default, see the "Advanced Reporting Configuration" section later in this chapter. You can use wildcards to reduce the number of values that is displayed when you click **Values**. Use the percent (%) symbol to substitute for any number of characters, the underscore (_) symbol to substitute for a single character, and the bracket ([ ]) symbols to search for literals. Although wildcards help reduce the number of values that is displayed when you click **Values**, you cannot use wildcards to reduce the number of results that is returned when you actually run a report by clicking **Display**. If you enter a wildcard and then click **Display**, the report searches for the wildcard as a literal value. For example, if you enter **%m%** when prompted for a computer name and then click **Display**, the report searches for computers that have the literal name %m%.

If the report does not have prompts, Report Viewer starts directly at the **Report Results** page for the selected report.

By default, a maximum of five reporting points appears on the **Run** menu and you can modify this number. For more information, see the "Advanced Reporting Configuration" section later in this chapter.

For performance reasons, Report Viewer limits the result set that is returned by a report query to 10,000 rows and you can modify this number. For more information, see the "Advanced Reporting Configuration" section later in this chapter.

The amount of time that is required to run a report depends on the amount of data that is returned by the report. With large reports, you might experience time-outs. If this happens, you can adjust the time-out settings. For more information, see the "Adjusting time-out settings" section later in this chapter.

For reports that are likely to return large amounts of data, such as status message reports or client installation reports, it is recommended that you create prompts or linked reports to limit the amount of data that is returned by any one report. By using prompts, a report can be limited to returning status messages only for a particular time period or to returning information about only clients in a specific site. For more information, see the "Report Prompts" and the "Report Links" sections earlier in this chapter.

Report Viewer cannot display different languages on a single reporting page. You can create individual reports that contain data in only one language.

Note

If double-byte character set (DBCS) information is not displayed correctly, such as Japanese computer names, you should configure Internet Explorer encoding to **Auto-Select**. Right-click anywhere in Report Viewer, point to **Encoding**, and then click **Auto-Select**. This overrides other encoding selections.

Using Report Data

When you run a report, there are several ways that you can use the report data in another application or offline. You can use the menu bar commands on the Report Results page to perform the following tasks:

- Print the report data.
- Copy the report data to the Clipboard.
- Display the report data as a chart (for reports configured to do so).
- Export the report data as a comma-delimited file (exporting report data is different from exporting report definitions).

Note

When you export report data, you only export the data that is contained in that report and not any of the data contained in the report's targets. For example, if you export a report that contains links to the Status Message page, you only export the status message IDs and not the actual data that is contained in the individual status messages.

- Add the report URL to your list of favorites.

Note

If you included any of the following characters in a report name, the characters are deleted from the favorite name when you add the report URL to your list of favorites: \ / : * ? " < > |

- Send the URL for the report by using e-mail (the recipient must have **Read** permission for the report and be a member of the SMS Reporting Users group to run the report).

Note

You should use the commands on the Report Result page menu bar to copy report data to the Clipboard or to print it. If you use the Internet Explorer shortcut menu or menu bar commands, you print or copy all elements on the page, rather than only the report data.

A report can return multiple result sets, for example, when you include more than one SELECT clause or a COMPUTE clause in an SQL statement. If a report is configured to display as a chart, and the report returns more than one result set, Report Viewer only displays the first result set as a chart. If you print a report that returns multiple result sets, copy it to the Clipboard, or export it to a comma-delimited file, all result sets are included.

You can sort the data within a result set by clicking a column heading. If the report has multiple result sets, you can sort the data in each result set independently. You can only sort by using one column at a time.

If a report has links to a target, link icons appear to the left of each row of data when you run the report in Report Viewer. When you click a link icon, the target opens in the same window. If a report has links to a target and returns multiple result sets, the same target is used for all result sets. For more information, see the "Report Links" section earlier in this chapter.

Using Predefined Reports

SMS 2003 provides a number of predefined reports. You can use these reports to gather a variety of useful information about your SMS site. You might find that you want to modify a predefined report to better meet your needs. If you modify the properties of a predefined report, you can no longer use the original report as designed. If you reinstall predefined reports, from an import or as part of a product upgrade, you lose your changes. To keep the original report intact, always make a copy of the predefined report, rename it, and then modify the new report to better meet your needs.

Note

When you run the predefined report called Computers that can be upgraded to WinXP, the report results correctly lists the operating system version for all Windows computers except those running Microsoft Windows 98. The caption for Microsoft Windows 98 computers reads **Microsoft Windows**.

Note

A number of predefined reports are designated to appear on the **Computer Details** page of Report Viewer. If you clear the **Display in computer details** check box, modify the SQL statement, or modify a report prompt for a predefined report, it might not work as intended. For more information, see the "Using the Computer Details Page" section later in this chapter.

Creating and Modifying Reports

Creating a new report or modifying a predefined report requires a working knowledge of SQL. For more information, see the "Creating and Modifying SQL Statements" section later in this chapter.

When you create a new report, you must specify a category. You can choose an existing category or create a new category. When you create a new category, it is added to the category list. Within a given category, report names must be unique. However, you can use duplicate report names in different categories. SMS 2003 assigns each new report a report ID number, which uniquely identifies the report. The category determines which tree branch the report appears in on the main page of Report Viewer.

You can configure a report to refresh its results automatically at a specified interval. This is especially useful for reports that you include in a dashboard or otherwise use to monitor information that changes frequently.

You can also configure a report to display its data as a chart. This is useful for reports that return counts, such as a report that provides a count of computers by network protocol. You can specify a chart title, a title and report column to use for the category (x) axis data, and a title and report column to use for the value (y) axis data. For the value (y) axis data, you should select a column that contains integer data. If you select a column that contains string data, some of the data might be truncated on the chart.

You can also specify a default chart type, such as a bar chart. A report user can choose to display the data with a different chart type. If a report returns multiple result sets, Report Viewer displays only the first result set as a chart. For more information about configuring display options for reports, see the SMS Help.

Note

The number of colors that a chart can display is limited to 16. If you have more than 16 items in a report, the colors are reused.

To display report data as a chart by using Report Viewer, you must have a licensed copy of Microsoft Office XP Web Components or Microsoft Office 2000 Web Components installed on the reporting point site system. You must also have a licensed copy of at least one Microsoft Office application installed on the reporting point site system. Office Web Components are installed with all Office XP editions and Office 2000 Professional, Premium, Developer, and Standard editions. They are not installed with Office 2000 Small Business or the stand-alone version of Microsoft Excel 2000.

To create or modify a report

1. In the SMS Administrator console, navigate to **Reports**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Reports
```

2. Right-click **Reports**, point to **New**, and then click **Report**.

   –Or–

   Right-click a report, and then click **Properties**.

3. Use the tabs in the **Report Properties** dialog box to configure the report properties.
   - Use the **General** tab to name the report, select a category, and create or modify the SQL statement. For more information about creating SQL statements, see the "Creating and Modifying SQL Statements" section later in this chapter.

> **Note**
> It is recommended that you select a category from the **Category** list. If you type a name in the **Category** box that does not match an existing category name exactly (case-sensitive), SMS creates a new category. New category names are added to the **Category** list.

   - Use the **Display** tab to configure the report to refresh automatically and to configure the report to display its data as a chart.
   - Use the **Links** tab to link the report to a target. For more information, see the "Report Links" section earlier in this chapter.
   - Use the **Security** tab to configure security options. For more information, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

For more information about configuring report properties, see the SMS Help.

► To clone (make a copy of) an existing report

1. In the SMS Administrator console, navigate to **Reports**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Reports
```

2. Right-click the report that you want to clone, point to **All Tasks**, and then click **Clone**.
3. In the **New report name** box, type a name for the new report, and then click **OK**.

> **Note**
> Because SMS creates a new report by using the same category as the report you are cloning, the name that you enter for the new report must be different than the name of the existing report.

Deleting Reports

When you delete a report, SMS removes the report object from the site database. The report no longer:

- Appears in the report list in the SMS Administrator console or Report Viewer.
- Appears in dashboards in which it was included.
- Is available as a target for other reports that contained links to it.

To delete a report

1. In the SMS Administrator console, navigate to **Reports**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Reports
```

2. Right-click the report that you want to delete, and then click **Delete**.

   The **Delete Report** dialog box displays the following information in the **Objects** list to alert you of the potential impacts of deleting the report:

   - Any dashboards that include the selected report.
   - Any reports that link to the selected report and for which you have **Read** permission.

Important

Reports for which you do not have **Read** permission are not displayed in the **Delete Report** dialog box. It is possible that deleting a report might impact reports other than the ones that are displayed.

Creating Report Prompts

A prompt is a report property that you can configure to request a parameter value from the user before running the report. When you include a prompt, the user is prompted to enter a parameter value prior to running the report. You can include more than one prompt in a report; however, each prompt must have a unique name.

Prompt names can only contain alphanumeric characters and must conform to the SQL rules for identifiers. For more information, see the SQL Server documentation.

To help report users enter parameter values, you can specify a default value when you create a prompt. You also can configure a prompt to display a list of valid values from which the user can choose. To do this, you create an SQL statement for the prompt, which is separate from the report's primary SQL statement. For example, if a report prompts the user for a computer name, and you want report users to be able to select from a list of names rather than typing one from memory, you can use an SQL statement. You must also allow for the use of wildcards to limit the number of values that is returned when you click the **Values** button for a prompted report. Use the variable *@_filterwildcard* to do so.

The following SQL statement returns a list of computer names:

```
begin
 if (@__filterwildcard = '')
 SELECT Name0 AS 'Computer Names' FROM v_R_SYSTEM ORDER BY Name0
 else
  SELECT Name0 AS 'Computer Names' FROM v_R_SYSTEM
  WHERE Name0 like @__filterwildcard
  ORDER BY Name0
end
```

> **Note**
> You should carefully create and test prompts that use an SQL statement to ensure that the statement does not return a large list of values, which can take a long time to run.

For more information, see the "Creating and Modifying SQL Statements" section later in this chapter.

To create or modify a report prompt

1. In the SMS Administrator console, navigate to **Reports**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Reports
```

2. Right-click **Reports**, point to **New**, and then click **Report**.

   –Or–

   Right-click a report, and then click **Properties**.

3. On the **General** tab, click **Edit SQL Statement**.
4. In the **Report SQL Statement** dialog box, click **Prompts**.
5. In the **Prompts** area, click **New** (gold star).
6. In the **Prompt Properties** dialog box, complete the following tasks:
   - In the **Name** box, type a name. You use this value as the prompt variable name to integrate the prompt into the SQL statement for the report. For more information, see the "Integrating Report Prompts" section later in this chapter.
   - In the **Prompt text** box, type the text that you want to appear as the display name for the prompt in Report Viewer. The prompt text informs the user about the type of value that is required for the prompt; for example, Computer name.
   - In the **Default value** box, type a value that you want to be automatically inserted into the prompt text box when a user runs the report. This is an optional setting and the report user can type in a different value.
   - To allow a report to run using an empty value for the prompt, select the **Allow an empty value** check box.

   > **Note**
   > If a report user leaves the value for a report prompt blank, and the report prompt is configured to allow an empty value, an empty string is used as the value when the report is run.

7. To use an SQL statement to retrieve a list of values from which the user can choose, select the **Provide a SQL statement** check box, and then click **Edit SQL statement**.
8. In the **SQL statement** box, enter a valid SQL statement for the prompt.

> **Note**
> A prompt SQL statement can return more than one column of values. However, when a report user selects an item from the list prior to running the report, only the value in the first column is returned to the prompt box.

For more information about creating an SQL statement, see the "Creating and Modifying SQL Statements" section later in this chapter.

Integrating Report Prompts

When you create a report prompt, it is not integrated automatically into the report's SQL statement. To integrate a prompt, you must specify the prompt name as a variable in the SQL statement of the report by using the syntax *@promptname*. The following SQL statement example includes a variable for a prompt that is named *prompt2*.

```
SELECT Sys.User_Name0 AS 'User Name', SYS.Name0 AS 'Comp Name' FROM v_R_SYSTEM
WHERE User_Name0 LIKE @prompt2
```

For more information, see the "SQL statement variables" section later in this chapter.

Exporting and Importing Reports

By using the Export Object Wizard, you can export one or more report objects. When you export report objects, SMS writes the object definitions to a MOF file. A MOF file is a text file that you can use to import report object instances into your SMS database. You can also use MOF files to import report object instances into another database. This can be useful for importing reports that you might download from the Internet or that are created by someone else and for exchanging reports between other SMS sites. Only the report object's definitions are exported, not any report data. When you import and run a report that was created at another SMS site, the report runs against your site database, not the original site database.

To export a report, you must have **Read** permission for the Reports security object class or instance. To import a report, you must have **Create** permission for the Reports security object class or instance.

When you export reports that have links, links to URLs are maintained; however, links to other targets are not. For example, if you export a report that links to another report, that link is not maintained and it must be manually reconfigured after the report object is imported.

The report ID is unique for each report. When you export a report, the report ID is not written to the MOF file. This prevents you from accidentally replacing an existing report by importing a MOF file in which a report ID for an imported report matches that of an existing report. When you import reports, SMS assigns each imported report a new report ID.

More than one report can have the same name, as long as each report is in a different report category. When you export reports, the report categories are written to the MOF file; however, the report categories do not appear in the Export Object Wizard. The unique report ID for each report does appear in the Export Object Wizard. To ensure that you are exporting the reports that you want, verify that the report ID of each report in the Export Object Wizard matches the report ID of each report as it appears in the details pane of the SMS Administrator console.

You can use the Export Object Wizard to export objects from only one object class (reports, collections, or queries) at a time. MOF files that are created by using the Export Object Wizard contain only one object class.

You can use the Import Object Wizard to import user-created MOF files that contain objects from multiple object classes. However, you must have **Create** permission for all object classes in a MOF file. Any objects for which you do not have permission are not imported. For example, if you import a MOF file that contains report and collection objects, but you have **Create** permission only for the Reports object class, the collection objects are not imported.

Note

To import a MOF file by using the Import Object Wizard, the file must be in Unicode file format. All MOF files that are exported by the Export Object Wizard are in Unicode file format.

To export report objects

1. In the SMS Administrator console, navigate to **Reports**, and then right-click **Reports**.

   –Or–

   In the SMS Administrator console, navigate to **Reports**, and then right-click a specific report that you want to export.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Reports
```

2. Point to **All Tasks**, and then click **Export Objects**.
3. Complete the Export Object Wizard, and then click **Finish**.

For more information about completing the Export Object Wizard, see the SMS Help.

Caution

When exporting reports, do not use a MOF file name that is the same as the existing MOF file name in the same folder. If you do, the data for the existing file is overwritten without warning.

To import report objects

1. In the SMS Administrator console, navigate to **Reports**, and then right-click **Reports**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Reports
```

2. Point to **All Tasks**, and then click **Import Objects**.
3. Complete the Import Object Wizard, and then click **Finish**.

For more information about completing the Import Object Wizard, see the SMS Help.

Caution

When importing reports, the properties of the existing report are overwritten without warning if you import a report with the same name and category as a report already in the database. To avoid this, open the MOF file by using Notepad or another text file application and review the object names against the names of existing objects in the SMS site database before importing the file.

Scheduling Reports

Report Viewer generates a unique URL for each report and dashboard that you run. The URL contains the report ID and the variable names that you used to run the report. You can use the URL to schedule a report or dashboard to run (or to run and export the data to a file) at a specified interval.

You do this by configuring the Scheduled Tasks feature of your operating system to start Internet Explorer with a URL.

To schedule a dashboard to run or a report to run and export to a file

1. Click the **Start** button, point to **All Programs**, point to **Accessories**, point to **System Tools**, and then click **Scheduled Tasks**.
2. Double-click **Add Scheduled Task**, and then click **Next**.
3. In the **Application** list, click **Internet Explorer**, and then click **Next**.
4. Enter a name for the task, select a time interval option, and then click **Next**.
5. Select the time and day that you want the task to start, and then click **Next**.
6. Enter a qualified user name and password, and then click **Next**.

7. Select the **Open advanced properties for this task when I click Finish** check box, and then click **Finish**.

8. To run and display a report at a specified interval, insert one space after the Internet Explorer command line in the **Run** box, and then type the URL of the report or dashboard. For example, **C:\PROGRA~1\INTERN~1\IEXPLORE.EXE http:\\ReportingPoint\SMSReporting_001\Report.asp?ReportID=15**

   –Or–

   To run, display, and export a report to a comma-delimited text file at a specified interval, insert a space after the Internet Explorer command line in the **Run** box, type the URL of the report or dashboard, and then type one of the following parameters immediately after the URL:

   - &ExportTo=<*Drive letter*>:\<*Path*>\<*Filename.txt*>, where *Drive letter* specifies a drive on the reporting point, not on the local computer. For example, **C:\PROGRA~1\INTERN~1\IEXPLORE.EXE http:\\Reporting_Point1\SMSReporting_001\Report.asp?ReportID=15& ExportTo=C:\ShareDrop\Report135.txt**
   - &ExportTo=\\<*Server name*>\<*Server share name*>\<*Filename.txt*>, where *Server name* and *Server share name* specify the reporting point and a share on that server. For example, **C:\PROGRA~1\INTERN~1\IEXPLORE.EXE http:\\Reporting_Point1\SMSReporting_001\Report.asp?ReportID=15& ExportTo=\\Server2\ShareDrop\Report135.csv**

Note

When you schedule a report to export to a comma-delimited text file, the Internet Explorer window remains open until you manually close it.

Using the Computer Details Page

The Computer Details page of Report Viewer displays a set of reports that have been designated to appear on that page. SMS 2003 provides a number of predefined reports, which appear on this page. You can also designate your own reports to appear on the **Computer Details** page. You can only designate reports that have one prompt or no prompts.

Reports that appear on the Computer Details page also appear in the list of reports on the main page of Report Viewer and in the SMS Administrator console. You can run the reports from these locations and from the Computer Details page.

To designate a report to appear on the Computer Details page

1. In the SMS Administrator console, right-click a report, and then click **Properties**.
2. In the **Report Properties** dialog box, click the **General** tab, and then select the **Display in Computer Details** check box.

   If you have **Read** permission for the report, it appears on the **Computer Details** page of Report Viewer.

> **Note**
>
> If you clear the **Display in Computer details report** check box of a predefined report or one that you have created, the report no longer appears on the **Computer Details** page of Report Viewer.

Many reports on the Computer Details page include a prompt that requests the user to enter a value before running the report. This value is usually a computer name but it can be a different value, such as a file name or a user name, depending on how the report is configured. When a user selects a report with a prompt on the Computer Details page, the title of the **Value** box changes to reflect the **Prompt Text** value that was specified when the prompt was created, such as Computer Name. The user can then enter a value and run the report. When a value is specified, the user can select other reports on the Computer Details page and run those reports by using the same value. For example, you might enter a computer name and run a report that provides operating system information about that computer. You can then run a report that provides processor information about the same computer.

If you link a report to the Computer Details page, that report must contain computer names (or other appropriate values) in one of its columns. A value from that column of the source report is automatically inserted into the **Value** box on the Computer Details page. For more information, see the "Report Links" section earlier in this chapter.

To use the Computer Details page

1. Open Report Viewer, and then navigate to the main page.
2. In the reports tree, click **Computer Details**.

   The **Computer Details** page appears in a separate window.
3. In the **Computer Details** reports tree, expand a category, and then click a report that you want to run.
4. In the **Values** box, type a value, and then press ENTER.

   The report appears in the right pane of the **Computer Details** page.

Using the Status Message Details Page

You can use the Status Message Details page to display information about a specific status message. The Status Message Details page is system-generated; you cannot modify or delete it.

The Status Message Details page displays the same information as the Status Message Viewer. You can use the Status Message Details page, instead of the SMS Administrator console, to integrate this status information into your reports.

A number of the predefined reports link to the Status Message Details page. You can also link reports that you create to the Status Message Details page. Any report that you link to the Status Message Details page must contain RecordID values in one of its columns. For more information, see the "Report Links" section earlier in this chapter.

Using Supplemental Reports

Supplemental reports are reports that you or others create outside of SMS and that you place in the Supplemental folder on a reporting point. If you have multiple reporting points, you must place a supplemental report on each of the reporting points from which you want users to access the report. Supplemental reports do not appear in the SMS Administrator console; they only appear in Report Viewer. However, the **Supplemental Reports** item does not appear in the Report Viewer tree until you install at least one supplemental report file on the reporting point.

Supplemental reports are not SMS database objects and therefore are not backed up routinely by the SMS backup service. You must back up these files manually. If you disable a reporting point, SMS does automatically back up any supplemental reports on that server to the root drive. For more information about how to locate and recover supplemental reports on a disabled reporting point, see the "Advanced Reporting Configuration" section later in this chapter.

Supplemental reports can be ASP files or any files that you can display by using Internet Explorer 5.0 or later, such as HTML files, Microsoft Office files, or text files.

You can run supplemental reports directly from Report Viewer or link other reports to a supplemental report by using the supplemental report's URL as a target. For more information, see the "Report Links" section earlier in this chapter.

To install a supplemental report file

1. On a reporting point site server, navigate to the following folder:

   <*Installation drive*>:\Inetpub\wwwroot\<*Reporting folder name*>\Supplemental

2. Place the supplemental report file in the Supplemental folder.

   You can now view and run the report by using Report Viewer. If Report Viewer is already started, you might need to refresh the view for the new report to appear.

Caution

Supplemental reports are not SMS database objects and are not backed up by the SMS backup service. You must back up these files manually.

► To run supplemental reports by using Report Viewer

1. On the Report Viewer main page, expand the **Supplemental Reports** item to view the list of supplemental reports.
2. Click the supplemental report that you want to run, and then click **Display**.

> **Note**
> The Supplemental Reports item appears only if you place at least one supplemental report in the designated folder on the reporting point.

Advanced Reporting Configuration

This section provides information about advanced configuration settings for reporting and contains the following topics:

- Changing the number of reporting points on the **Run** menu
- Adjusting time-out settings
- Changing the number of rows returned by a report query
- Changing the number of values returned by clicking **Values**
- Locating supplemental reports on a disabled reporting point

Changing the number of reporting points on the Run menu

By default, a maximum of five reporting points appears on the **Run** menu. You can modify the number of reporting points that appear the **Run** menu by using the following procedure.

► To change the number of reporting points on the Run menu

1. On the computer on which the SMS Administrator console is installed, run Regedt32.exe or Regedit.exe.
2. Navigate to the following registry key:

   \HKEY_LOCAL_MACHINE\Software\Microsoft
3. Under the **Microsoft** key, create three new keys that result in the following structure

   \HKEY_LOCAL_MACHINE\Software\Microsoft\SMS\Admin UI\Reporting
4. In the **Reporting** key, create a **DWORD** value named **MenuCount**, and then type a value.

   The value is the maximum number of reporting points that can appear on the **Run** menu.

> **Note**
> To have all available reporting points appear on the Run submenu, type 0 (zero) as the DWORD value.

Adjusting time-out settings

When you run a report, Report Viewer uses ASP files that are stored on a reporting point. The IIS default setting for the ASP script time-out is 90 seconds. This should be sufficient for running reports in most environments. However, in certain situations some reports might time out before finishing, such as those with a slow network connection, insufficient hardware, or an SQL statement that is inefficient or returns a large set of records. If you receive error messages indicating that the maximum time to run a script has been exceeded, you might need to increase the script time-out setting.

Note

The script time-out setting must not be less than either of the following control time-out settings, which are described later in this section:

- Session("DBConnectionTimeout")
- Session("DBCommandTimeout")

For information about how to increase the ASP script time-out setting, see article number 268364 in the Microsoft Knowledge Base at http://support.microsoft.com.

To retrieve data from the SQL Server views in the SMS site database, the ASP script calls an ActiveX control. In the call, the script passes two time-out settings as parameters:

Session("DBConnectionTimeout") This setting specifies the number of seconds within which the ActiveX control must connect to the SMS site database server. The default is 30 seconds.

Session("DBCommandTimeout") This setting specifies the number of seconds within which the ActiveX control must receive data back from the SMS site database server. The default is 60 seconds.

If you experience time-outs when running reports, you might need to increase these time-out settings in addition to increasing the ASP script time-out setting.

The time-out settings are specified in the Global.asa file. You can open and modify this file to increase the settings by using Notepad or another a text editor. If you have multiple reporting points, you need to modify the Global.asa file on each of the reporting points on which you are experiencing time-outs. The Global.asa file is located in the following folder: <*Installation drive*>:\Inetpub\wwwroot\<*Reporting folder name*>\.

Time-outs can also impact the performance of dashboards. When one or more reports contained in a dashboard experience time-outs, time-out error messages might appear in some cells and other cells might not display data at all. You should carefully set time-outs and report refresh intervals so that reports that are used in dashboards do not time out or refresh before the dashboard can display all reports.

Changing the number of rows returned by a report query

For performance reasons, Report Viewer limits the result set returned by a report query to 10,000 rows. You can modify the registry to override this limit and return any number of rows by using the following procedure.

To change the number of rows returned by a report query

1. On all computers on which a reporting point is enabled, run Regedt32.exe or Regedit.exe.
2. Navigate to the following registry key:

   \HKEY_LOCAL_MACHINE_USER\Software\Microsoft\ SMS\Reporting
3. In the **Reporting** key, create a **DWORD** value named **Rowcount**, and then set its value to the number of rows that you want returned. If you want to return all rows, set the value to 0xffffffff, which is the hexadecimal equivalent of –1.

   The appropriate number of rows is returned by any report query that is run from this reporting point.

> **Note**
>
> If you set Rowcount to a number that is not valid (such as 0 or a number less than –1), Report Viewer returns the default maximum of 10,000 rows.

Changing the number of values returned by clicking Values

For performance reasons, Report Viewer limits the number of values returned when you click **Values** in a prompted query to 1,000 rows. You can modify the registry to override this limit and return any number of values by using the following procedure.

To change the number of values returned by clicking Values

1. On all computers on which a reporting point is enabled, run Regedt32.exe or Regedit.exe.
2. Navigate to the following registry key:

   \HKEY_LOCAL_MACHINE_USER\Software\Microsoft\ SMS\Reporting
3. In the **Reporting** key, create a **DWORD** value named **Values Rowcount**, and then set its value to the number of values that you want returned. If you want to return all values, set the value to 0xffffffff, which is the hexadecimal equivalent of –1.

Locating supplemental reports on a disabled reporting point

Supplemental reports are reports that you or others create outside of SMS 2003 and that you place in a designated folder of a reporting point. Supplemental reports are not SMS database objects and therefore are not backed up routinely by the SMS backup service. You must back up these files manually.

If you disable the reporting point, SMS does automatically back up any supplemental reports on that server to a folder on the root drive. If you enable the reporting point again, SMS automatically moves the supplemental reports from the backup directory to the designated folder on the reporting point.

To locate supplemental reports on a disabled reporting point

1. On a computer on which a reporting point is disabled, run Regedt32.exe or Regedit.exe.
2. Navigate to the following registry key:

   \HKEY_LOCAL_MACHINE\Software\Microsoft\ SMS\Client\BackupSuppRptDir

   The value for the **BackupSuppRptDir** key is the path of the directory that SMS placed the supplemental reports.

Creating and Modifying SQL Statements

The principal element of a report is its SQL statement. The SQL statement determines which records and fields are returned each time that a user runs the report. The SQL statement accesses read-only SQL Server views, rather than your SMS site database tables. The reporting interface supports most SQL keywords and clauses that can be used for the read-only views. The primary clause that is used for creating SQL statements is the SELECT clause.

You can also create SQL statements to use for a report prompt. When a user runs a report with an SQL statement for a prompt, the SQL statement returns a list of values from which the user can choose. You can create reports prompts that do not use an SQL statement. For more information, see the "Report Prompts" section earlier in this chapter.

To create your own reports requires a working knowledge of SQL. It is not within the scope of this chapter to teach you SQL. However, this section does provide information about how the reporting interface can help you create SQL statements. Although the interface can help you, it does not automatically create complete SQL statements, nor does it validate them. However, SMS 2003 does perform limited syntax checks of the SQL statement.

You can use Microsoft SQL Server SQL Query Analyzer or another SQL query builder to create SQL statements, and then copy and paste the statements into reports. This might be helpful if you want to create longer or more complex statements.

Important

You must write case-sensitive queries for reports when they will be run against a case-sensitive SQL Server. Otherwise, the report will not run correctly and the SQL Server will generate errors.

To create an SQL statement, you need an understanding of the SQL Server views that expose data from your SMS site database. Before creating SQL statements, see the "SQL Server Views" section later in this chapter.

The process for creating or modifying an SQL statement in a report is the same.

To create or modify an SQL statement for a report

1. In the SMS Administrator console, right-click **Reports**, point to **New**, and then click **Report**.

   –Or–

   In the SMS Administrator console, right-click a report and click **Properties**.
2. On the **General** tab, click **Edit SQL Statement**.
3. In the **SQL statement** box, enter a valid SQL statement.

To create or modify an SQL statement for a prompt

1. In the SMS Administrator console, right-click **Reports**, point to **New**, and then click **Report**.

   –Or–

   In the SMS Administrator console, right-click a report, and then click **Properties**.
2. On the **General** tab, click **Edit SQL Statement**.
3. In the **Report SQL Statement** dialog box, click **Prompts**.
4. In the **Prompts** dialog box, click **New** (gold star).

   – Or –

   In the **Prompts** dialog box, click a prompt, and then click **Properties**.
5. In the **Prompt Properties** dialog box, select the **Provide a SQL statement** check box, and then click **Edit SQL Statement**.
6. In the **SQL statement** box, enter a valid SQL statement for the prompt.

> **Note**
> If you modify or delete a prompt in a report, links to that report from other reports might be broken. This includes modifying an SQL statement that is used for a prompt.

Building an SQL Statement

The reporting interface has features that can help you build SQL statements for reports that run against the SQL Server views.

> **Note**
> While the features of the **Report SQL Statement** dialog box can assist you in building an SQL statement, the interface does not automatically create complete SQL statements, nor does it validate them. However, SMS 2003 does perform limited syntax checks of the SQL statement.

When you initially open the **Report SQL Statement** dialog box for a new report, the **SQL statement** box contains the following sample SQL statement:

```
SELECT * FROM V_R_System where V_R_System.Name0 = 'computer_name'
```

A SELECT statement specifies the columns to be returned by the statement. It retrieves the data from the SQL Server views and presents it to the user in one or more result sets.

Note

SQL statements are not case-sensitive.

You can leave the asterisk (*) that follows the SELECT keyword to return all columns or replace it with the specific column names that you want the report to return (for example, **User_Domain0** or **User_Name0**). The FROM clause indicates the SQL Server view from which the data is retrieved and always follows the SELECT keyword. For more information, see the "SQL Server Views" section earlier in this chapter.

You can create multiple SELECT statements within an SQL statement for a report, which returns multiple result sets. The following is an example:

```
SELECT * FROM v_StatMsgModuleNames
SELECT * FROM v_SoftwareProduct
```

Note

If you use multiple SELECT statements for a report, you should test each statement individually to ensure that it runs successfully. When a report fails, it returns an error code indicating the failure. When you use multiple SELECT statements, they are treated as a single request; if any statement fails, only one error code is returned and the report fails.

The **Report SQL Statement** dialog box has controls that you can use to help you build SQL statements. You can use the **Views** and **Columns** lists to insert view and column names and the **Values** button to insert column values into the SQL statement.

Note

The **Report SQL Statement** dialog box controls insert data in the SQL statement at the position of the cursor. When you first open the **Report SQL Statement** dialog box, the cursor is positioned at the beginning of the statement. You should position the cursor before inserting data.

To insert a view name

1. In the **SQL statement** box, position your cursor in the SQL statement where you want to insert a view name.
2. In the **Views** list, click a view name, and then click **Insert**.

To insert a column name

1. In the **SQL statement** box, position the cursor in the SQL statement where you want to insert a column name.
2. In the **Views** list, select the view that contains the column or columns that you want to add.
3. In the **Columns** list, click a column name, and then click **Insert**.

To insert a column value

1. In the **SQL statement** box, position your cursor in the SQL statement where you want to insert a column value.
2. In the **Columns** list, select a column, and then click **Values**.
3. In the **Values shown** area, click the **Previous** and **Next** buttons to scroll through the values.
4. To apply a filter to limit the number of values that is returned, click **Set**.
5. In the **Set Filter** dialog box, specify the filter criterion, and then click **OK**.
6. In the **Values** list, click the value that you want to add, and then click **OK**.

For more information about using the **Report SQL Statement** dialog box, see SMS Help.

SQL keywords and clauses

The following are some other commonly used SQL keywords and clauses that you might find helpful for creating reports:

AS Specifies an alias for a column name. An alias replaces the column display name in the result set. Therefore, when displaying the result set, Report Viewer uses aliases as the column headings, rather than the default column display names. You can uses aliases to create column headings that might be more understandable to report users. In the following example, User_Name0 is assigned the display name *User Name*. You can also use an alias in place of the column name in an ORDER BY clause, but not in a WHERE clause.

WHERE Specifies a search condition that restricts the rows that are returned. This condition can be based on a specified value from one of the selected columns, as compared to a variable or a string.

ORDER BY Specifies that the result set be sorted in ascending sequence based on the value in a specified column. In the following example, the SQL statement sorts the result set by data in the column User Name, and then by the data in the column Comp Name.

COMPUTE Generates totals that appear as additional summary columns at the end of the result set. Using a COMPUTE clause returns a report with multiple result sets.

The following sample statement provides examples of these keywords and clauses.

```
SELECT User•Name0 AS 'User Name', Name0 AS 'Comp Name' FROM v_R_System
WHERE User•Name0 LIKE @variable2
ORDER BY User•Name0, Name0
COMPUTE COUNT (User•Name0) BY User•Name0, Name0
```

SQL statement variables

You use variables to integrate report prompts into the SQL statement for a report. Report prompts provide a means for the user to enter a dynamic value each time that the user runs a report. Report Viewer uses that value as a variable value in the SQL statement to target or limit the data that is returned. For example, in a report that returns data about a client, the user might be prompted to enter a computer name.

When you create a report prompt, you assign it a prompt name. To integrate the prompt into the SQL statement, you define the prompt name as a variable at the appropriate place in your SQL statement. You can create more than one prompt, each with its own variable; however, the name for each prompt must be unique within a report. For more information, see the "Integrating Report Prompts" section earlier in this chapter.

Converting Coordinated Universal Time (Greenwich Mean Time) to local time

By default, time data is stored in the SMS database in the local time of the system that generated the data. However, some time data is stored in Coordinated Universal Time, specifically status messages stored in the v_StatusMessage and v_ClientAdvertisementStatus views and in the software metering data and summarization views. In addition, some time data might be stored in Coordinated Universal Time, depending on which time format that you selected when creating the data, such as the ExpirationTime in the v_Advertisements view.

When you create an SQL statement for a report that includes a column with Coordinated Universal Time data, the data appears in the report in Coordinated Universal Time. If you prefer to have local time appear in the report, you can use the implicit variable @__timezoneofffset in your SQL statement. When you use this variable, SMS returns the offset from Coordinated Universal Time in seconds. To convert to local time, use the following syntax:

```
DATEADD(ss,@__timezoneoffset,< time column name>).
```

SQL statement examples

The following examples show how to use the SQL Server views to create useful SQL statements for reports:

- To return the list of all available views, which can be helpful for creating other reports, use the following statement:

```
SELECT Type, ViewName AS 'View Name' FROM v_SchemaViews
```

- To return the list of available inventory views, use the following statement:

```
SELECT Type, ViewName AS 'View Name' FROM v_SchemaViews WHERE
Type='Inventory'
```

- To return the display name of resources based on the resource type number (5 = System), use the following statement:

```
SELECT DisplayName AS 'Display Name' FROM v_ResourceMap WHERE ResourceType=5
```

- To determine discovery properties for a particular resource type, use the following statement:

```
SELECT * FROM v_ResourceAttributeMap WHERE ResourceType=5
```

- To list the inventory groups for a particular resource type, use the following statement:

```
SELECT InvClassName FROM v_GroupMap WHERE ResourceType=5
```

SQL Server Views

Your SMS site database contains a large collection of information about your networks, computers, users, user groups, and many other components of your computing environment. The information is often very detailed. The SMS site database also contains objects that represent familiar SMS items, such as advertisements, packages, queries, reports, and status messages.

Reporting uses SQL Server views that mirror the SMS site database schema structure that is created by the SMS Provider in Windows Management Instrumentation (WMI). The SQL Server views provide access to data from tables in the SMS site database, which is stored in SQL Server. Using views offers a faster and more efficient reporting option over accessing the data by using the SMS Provider. The SMS Provider is the application that communicates between WMI and the SMS site database.

When you use the reporting interface to create a report, you can browse the views, their columns, and their values and use them to create SQL statements. For more information, see the "Creating and Modifying SQL Statements" section earlier in this chapter.

You might find that some objects and properties are not initially present in your SMS site database or in the corresponding tables. Some are created as the result of a particular discovery method. Some hardware and software classes are not collected by default but must be enabled. For more information about SMS object classes, attributes, and properties, and which are initially enabled, see the *Microsoft Systems Management Server 2003 Software Development Kit*.

The SMS SDK is an excellent source of information about the SMS database and its object classes and attributes. You can download the SMS SDK from the Microsoft Web site at http://www.microsoft.com/smserver/downloads.

Another way to understand the SMS classes is to browse the underlying WMI classes, as described in Appendix B, "Windows Management Instrumentation."

Views Setup

During setup, SMS 2003 creates two types of SQL Server views:

Static SMS 2003 creates these views with data from static (unchanging) tables by running a Create View script.

Dynamic SMS 2003 creates these views with data from tables with a dynamic (changing) schema by running stored procedures that are installed during setup.

Discovery, inventory, and collection views fall into the dynamic category, where new tables or columns might be added during the operation of your SMS site. For example, if you create a new collection or programmatically modify the inventory information that SMS 2003 collects from clients, the views change as well. The views refresh automatically anytime that the schema of the underlying tables change.

When you extend the discovery or inventory classes, the data stays in the SMS site database, unless you run a tool to remove it. Views related to individual collections are removed if the collection is removed. If a collection view is removed, any reports that run against it no longer return results.

View Nomenclature

Because the SQL Server views schema conforms to the corresponding WMI schema, the views closely align with WMI resource classes. SMS object types are WMI classes, and SMS attributes are WMI properties.

The names of the SQL Server views are designed to closely resemble the SMS Provider WMI schema. Because the view names and view column names must be valid SQL identifiers, there are some differences between WMI and SQL Server view names. To ensure uniqueness with built-in SQL Server syntax, the column names in the inventory and discovery views end with a zero. Although there are exceptions, this is the main difference between WMI property names and the corresponding column names for the inventory and discovery views.

In most cases, the following rules are applied to convert WMI object names to their corresponding SQL Server view names:

- The beginning of each view name is changed from SMS_ to v_.
- Object names in the view schema are limited to 30 characters, which ensures compatibility with earlier SQL Server versions. Object names longer than 30 characters are truncated.
- Column names for views other than inventory or discovery are the same as the WMI property names.

For example, the WMI class **Win32_DisplayControllerConfiguration** is represented in the SMS Provider WMI schema as the **SMS_G_System_Display_Controller_Configuration** attribute class. The name of the view that exposes this table of attribute-class data, truncated to 30 characters, is v_GS_Display_Controller_Confi, with G_System truncated to GS.

Discovery views

Discovery data views consist of system resource objects (systems, users, user groups), which include any resources that were discovered on the network by a variety of means. The type of information that SMS gathers depends on the type of resource that is discovered. For example, some resources, such as printers, might not have the **Operating system name and version** property.

In the SMS Provider WMI schema, the SMS_R_System table contains discovery information for all SMS resources. The views for discovery data differ from their WMI counterparts in that the array properties (such as **IPAddresses**) are represented as separate views from the scalar properties (such as **Resource_Domain**). For example, with the WMI **System Resource** class (the **SMS_R_System** class), the scalar properties are contained in the v_R_System view. The array values are contained in the view tables that begin with v_RA, such as the v_RA_System_IPAddresses and v_RA_System_MACAddresses views. For more information, see Table 11.2.

Each view for an array property consists of two columns:

- A column that contains the data
- ResourceID, which links the tables

For example, for the v_RA_System_IPAddresses view, the data column is **IPAddresses0**.

Inventory data views

Inventory data views contain hardware and software inventory information about the clients in your SMS hierarchy. SMS collects inventory data when you enable the Hardware Inventory Client Agent or the Software Inventory Client Agent.

During the initial hardware inventory, by default, SMS collects as many as 200 hardware properties, which include details such as the:

- Boot configuration settings.
- BIOS settings.
- Number of disk drives.
- Type of processor.
- Amount of memory.
- Operating system.
- Monitor and display settings.
- Computer name and IP address.
- Network adapters.

In the SMS Provider WMI schema, the SMS_G_System tables contain inventory information for all SMS resources. The **ResourceID** field links these tables to the SMS_R_System table, which contains discovery information for the same resources. The current inventory data is represented by views that begin with v_GS; for example, v_GS_Modem_Device or v_GS_Processor. The history inventory data is represented by the views that begin with v_HS; for example, v_HS_Modem_Device. For more information, see Table 11.2.

There are also two inventory views for special use:

v_GS_System A subset of the discovery data, which contains information about clients, such as domain, name, and system type.

v_GS_Workstation Contains information about when inventory was last collected on a client.

Table 11.2 describes nomenclature for the SMS discovery and inventory classes and their SQL Server view equivalents.

Table 11.2 Nomenclature for Views

Class	SMS class	SQL Server views
Discovery:		
Scalar class	SMS_R_<*resource type name*>	v_R_<*resource type name*>
Array class	No separate classes for arrays	v_RA_<*resource type name*>_<*property name*> [1]
Inventory:		
Current inventory classes	SMS_G_System_Current_<*group name*>	v_GS_<*group name*> [2]
History inventory classes	SMS_G_System_History_<*group name*>	v_HS_<*group name*>
Extended history classes	SMS_GEH_<*group name*>	No equivalent view [3]
Custom Resource Inventory:		
Current inventory classes	SMS_G_<*resource type name*>_<*group name*>	v_G_<*resource type number*>_<*group name*> [4]
History inventory classes	SMS_GH_<*resource type number*>_<*group name*>	v_H_<*resource type number*>_<*group name*>

1. For example, **v_RA_System_IPAddresses** or **v_RA_User_GroupName**. For more information, see the "Discovery views" section earlier in this chapter.

2. For example, **v_GS_Modem_Device** or **v_GS_SoftwareFile**.

3. There is no equivalent view for the Extended History classes because these are implemented as a stored procedure. The extended history inventory class stores incremental changes to inventory objects, both current and obsolete.

4. For example, **v_G_6_VendorData**. In this example, it is assumed that a new resource type, such as Vending Machine, was added to the system and assigned the resource type number 6 and that inventory groups were added. You can associate the resource type number with the resource type name and its group classes by using the schema information views. For more information, see the "Schema information views" section later in this chapter.

Schema information views

Schema information views provide information about the available views and the schema for the inventory and discovery classes. These views are particularly useful for determining the names of inventory views for custom resource types.

Table 11.3 describes the data in the schema information views.

Table 11.3 Schema Information Views

View	Data
v_SchemaViews	All views in the view schema family
v_ResourceMap	All discovery resource type views
v_ResourceAttributeMap	Attributes for each resource type
v_GroupMap	Inventory groups for each inventory architecture
v_GroupAttributeMap	Attributes for each inventory group
v_ReportViewSchema	All the classes and properties

Collection views

Each collection in the SMS Administrator console is represented by its own view, which includes data about each resource that is a member of the collection. Collection view names begin with v_CM_RES_COLL and end with the unique collection ID number. For example, the All Systems collection is represented by the v_CM_RES_COLL_SMS0001 view. When you create a new collection, SMS 2003 automatically creates a new view to represent the collection.

In addition to the views for individual collections, there are views that contain data about the collection object instances in the collection class.

Table 11.4 describes the collection object views.

Table 11.4 Collection Object Views

View	Data
v_Collection	Lists all collections, with data such as when the membership was last refreshed
v_CollectToSubCollect	Associates a parent collection with its subcollections by collection ID
v_FullCollectionMembership	Lists the members of all collections
v_CollectionRuleDirect	Identifies the resource type and ID for collections with direct membership rules
v_CollectionRuleQuery	Identifies the query for collections with query-based membership rules

Status views

Status messages are generated by SMS components and represent the flow of activity within an SMS site and hierarchy. The status messages can provide valuable information that you can use to assess the health of your SMS system. There are several views that contain information about status messages such as component name, message ID, module name, message type, severity, time, site code, and computer name.

Status message instances consist of properties that are stored in the database, which are represented primarily by the v_StatusMessage view, and message strings stored in dynamic-link library (DLL) files. When you view a message by using the SMS Administrator console, Status Message Viewer, or the **Status Message Details** page in Report Viewer, SMS creates the instance of status messages by combining the various parts.

The v_StatMsgModuleNames view associates module names, such as SMS Client or SMS Provider, to the corresponding DLL file name, such as Climsgs.dll or Provmsgs.dll. The v_StatmsgInsStrings view contains information that SMS inserts into standard status messages, such as component or site names.

Status summarizers produce summaries from status messages and other data in the SMS site database. Status summaries are produced in real time as the summarizers receive status messages from SMS components. You can use status summarizers to view a snapshot of the status and health of the site systems, components, packages, and advertisements in your site.

Data in a status summary is classified as either a count or a state. A *count* is a tally of events that occurs over a specific period of time, such as the number of error status messages reported by SMS Executive since the beginning of the week. A *state* is the last known condition of something, such as the number of free bytes that is available for the SMS site database.

Each of the status summaries contains some state data. Only the Component Status and Advertisement Status summaries contain count data. The status summarizer views contain data such as the number of information, warning, and error messages for a site within a specified interval or the state of all components in a site at a specified internal.

Other views

In addition to the views described earlier in this chapter, there are views that contain information about a variety of SMS objects. As with the individual inventory views, the names of the views for these objects are designed to be self-explanatory. The following list briefly describes the types of information that you can obtain from these views:

Advertisements These views contain information such as package ID, collection ID, time package was presented, and time that the advertisement expires.

Packages This view contains information such as package ID, version, manufacturer, priority, and preferred address type.

Queries This view contains information such as name, expression (the WQL query text), query ID, object type targeted by the query, and collection ID to which the query is limited (if applicable).

Reports These views contain information about reports such as name, comment, category, SQL statement, and links. There are also several views that contain data about dashboards. These views contain information such as name, number of columns and rows, and which reports each dashboard contains.

Sites This view contains information about your SMS site such as server name, site code, SMS version and build numbers, type, and status.

Security These views contain security information about permissions that are granted to users and user groups to perform operations on secured SMS object classes and instances, such as collections, packages, and reports.

Working with Dashboards

A dashboard is a set of reports in a grid that you can display within a single window of Report Viewer. You can use dashboards to quickly obtain overview information about a variety of topics. You can copy a predefined dashboard and modify it to meet your needs or create your own custom dashboards. You cannot export or import a dashboard.

Creating and Managing Dashboards

You use the SMS Administrator console to create and manage dashboards. You can view and navigate the list of dashboards either in the SMS Administrator console or in Report Viewer. You run dashboards by using Report Viewer.

To include a report in a dashboard, you must have **Read** permission for the Reports security object class or the report instance. Dashboard users must also have **Read** permission for the Reports security object class or instances to view the results of reports included in a dashboard. For more information about permissions, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Because you cannot configure a dashboard to pass prompt values to a report that it contains, you can only include reports that do not require prompts. For more information about prompts, see the "Report Prompts" section earlier in this chapter. You can include reports that have links. For more information about report links and targets, see the "Report Links" section earlier in this chapter.

The following sections describe how to perform dashboard-related tasks:

- Viewing the List of Dashboards
- Running Dashboards
- Using Dashboard Data
- Scheduling Dashboards
- Creating, Configuring, and Managing Dashboards

Viewing the List of Dashboards

You can view and navigate the list of dashboards by using either the SMS Administrator console or Report Viewer.

Note

Because dashboards are not secured objects, all users can view the list of dashboards. However, reports that are contained in a dashboard might be secured and cannot be viewed unless the user has **Read** permission.

► To view the list of dashboards by using the SMS Administrator console

- In the SMS Administrator console, navigate to **Dashboards**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Dashboards
```

The list of dashboards appears in the details pane.

In the SMS Administrator console, you can sort dashboards by name or by dashboard ID. To sort the list of dashboards, click the appropriate column heading.

► To view the list of dashboards by using Report Viewer

1. In the SMS Administrator console, navigate to **Dashboards**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Dashboards
```

2. Right-click **Dashboards**, point to **All Tasks**, and then point to **Run**.
3. On the **Run** menu, click the name of the reporting point that you want to use to start Report Viewer.

   The list of dashboards appears under **Dashboards** on the Report Viewer main page.

Note

You can also start Report Viewer by directing Internet Explorer to the URL that is specified for a reporting point.

Running Dashboards

You run dashboards by using Report Viewer. You can start Report Viewer to run a dashboard from the SMS Administrator console or by entering the dashboard's unique URL in the **Address** box of Internet Explorer.

You can also use the URL to schedule dashboards to run automatically at a specified time. The steps for doing this are the same as those for scheduling reports. For more information, see the "Scheduling Reports" section earlier in this chapter.

Note

When one or more reports contained in a dashboard experience time-outs, time-out error messages might appear in some cells and other cells might not display data at all. For more information, see the "Adjusting time-out settings" section earlier in this chapter.

► To run dashboards by using the SMS Administrator console

1. In the SMS Administrator console, navigate to **Dashboards**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Dashboards
```

2. Right-click **Dashboards**, point to **All Tasks**, and then point to **Run**.
3. On the **Run** submenu, click the name of the reporting point that you want to use to start Report Viewer.

   The list of dashboards appears under **Dashboards** on the Report Viewer main page.
4. Select a dashboard, and then click **Display**.

Using Dashboard Data

When you have run a dashboard, you can:

- Print it.
- Add the dashboard to your list of favorites.
- Open the individual reports in a separate window.
- Open a target of an individual report in a separate window.

Each report displayed in a dashboard has a link icon on the left side of the title bar. To display the individual report in a separate window of Report Viewer, click the icon. If a dashboard displays a report that has links, you can also click the link icons in that report to display the target in a separate window of Report Viewer. For more information about report links, see the "Report Links" section earlier in this chapter.

You can configure individual reports to refresh automatically at a regular interval. This feature can be especially helpful for reports that you include in a dashboard. For more information about configuring reports to refresh automatically, see the "Creating and Modifying Reports" section earlier in this chapter.

Scheduling Dashboards

SMS generates a unique URL for each report and dashboard. You can use the URL to schedule a report or dashboard to run (or to run and export to a specified file location) at a specified interval.

You can do this by configuring the Scheduled Tasks feature of your operating system to start Internet Explorer with a URL. For more information, see the "Scheduling Reports" section earlier in this chapter.

Creating, Modifying, and Deleting Dashboards

When you create a new dashboard, you determine the number of reports that it can contain by specifying the number of rows and columns. You can limit the height of the cells in which the reports display to minimize the size of the dashboard window. The default height for each report cell is 250 pixels.

When you define the number of cells in a dashboard, you can select the reports that you want to display in the cells.

To create a new dashboard

1. In the SMS Administrator console, navigate to **Dashboards**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Dashboards
```

2. Right-click **Dashboards**, point to **New**, and then click **Dashboard**.
3. Click the **General** tab, and then enter a dashboard name, a comment, and the cell height.
4. Click the **Reports** tab, and then set the number of rows and columns, specify reports for the cells, and adjust the order of the reports.

> **Note**
> You cannot add a report that requires a prompt to a dashboard.

For more information about configuring the dashboard properties, see SMS Help.

To clone an existing dashboard

1. In the SMS Administrator console, navigate to **Dashboards**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Dashboards
```

2. Right-click the dashboard that you want to clone, point to **All Tasks**, and then click **Clone**.
3. In the **New dashboard name** box, type a name for the new dashboard, and then click **OK**.

To modify a dashboard

1. In the SMS Administrator console, navigate to **Dashboards**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Dashboards
```

2. Right-click the dashboard that you want to modify, and then click **Properties**.
3. Click the **General** tab, and then modify the settings as needed.
4. Click the **Reports** tab, and then modify the settings as needed.

For more information about configuring the dashboard properties, see the SMS Help.

To delete a dashboard

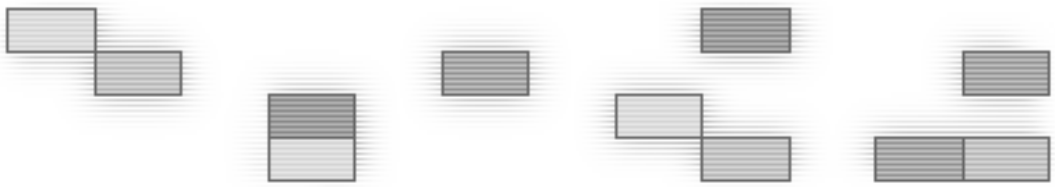
1. In the SMS Administrator console, navigate to **Dashboards**.

```
Systems Management Server
 └ Site Database (site code-site name)
  └ Reporting
   └ Dashboards
```

2. Right-click the dashboard that you want to delete, and then click **Delete**.

PART 3

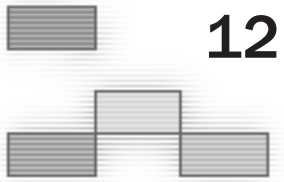
Maintaining SMS in Your Organization



This part of the *Microsoft Systems Management Server 2003 Operations Guide* guides you through the tasks that are required to maintain your Systems Management Server 2003 sites.

CHAPTER 12

Determining Product Compliance



Microsoft® Systems Management Server (SMS) 2003 provides functionality that helps you analyze and maintain product compliance on SMS client computers in your organization.

Organizations might set guidelines and standards for client software and require that clients follow these rules. You can enable software inventory or software metering and then use product compliance with these rules to detect clients that are running noncompliant software. You can then use software distribution to bring the software into compliance.

In This Chapter

- Using SMS for Product Compliance
- Customizing Product Compliance Data

To benefit most from this chapter, it is important that you are familiar with the overview of product compliance in Chapter 3, "Understanding SMS Features," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

This chapter does not cover product compliance issues that are related to upgrading to SMS 2003. For information about upgrade issues, see Chapter 14, "Upgrading to SMS 2003," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Using SMS for Product Compliance

To maintain product compliance in your organization, you must first determine if there are compliance issues by analyzing data in your organization. By using the analysis results, you can determine if there are any issues that need to be resolved.

To use the product compliance feature, you must generate product compliance data. Product compliance data is a collection of the software guidelines and standards that are set in your organization and organized and stored in a specific way. Every compliance guideline must include two key data items: compliance type and compliance level.

Compliance type A type of product guideline or standard. You can create as many compliance types as required in your organization.

Compliance level A descriptive measure of the level of product compliance with respect to a specific compliance type.

For example, your organization might set a requirement to use only the latest version of Microsoft Office. To detect compliance issues with this standard, you can define an Office Standard compliance type. For that compliance type, you might define the following compliance levels: compliant, noncompliant, and compliant with issues — requires a patch. Using these definitions, you can create the following product guidelines:

- Microsoft Office XP is compliant with the Office Standard compliance type.
- Microsoft Office 2000 is compliant with issues — requires a patch with the Office Standard compliance type.
- Microsoft Office 97 is noncompliant with the Office Standard compliance type.

In a similar way, you can define other product guidelines to generate additional product compliance data. You can then run queries and reports that compare product compliance data against software inventory data or software metering data to determine which clients are noncompliant.

To resolve compliance issues, you can use the software distribution feature to upgrade software or to add specific patches to bring software into compliance. You might need to remove noncompliant software that cannot be updated.

Compliance Analysis

This section describes the administrative tasks that are involved in using the product compliance feature.

To bring software into compliance, you must analyze product compliance and detect any issues that need to be resolved. Part of the analysis process is ensuring that the SMS site database contains the following required data:

- Software inventory data
- Product compliance data
- Queries and reports that analyze compliance

To analyze product compliance in your organization

1. Define compliance types according to software guidelines and standards that are set in your organization.
2. For each compliance type, define associated compliance levels.
3. Sort and categorize the software guidelines and standards according to the compliance types and compliance levels that you specified.

   Match the products that clients are most likely to use to the specified compliance types. For each product in the list, choose the appropriate compliance level. Use the compliance level list that is associated with the product's compliance type.
4. Add the data as product compliance records to the SMS site database. There are two methods to add product compliance data to the SMS site database:
   - Manually, by entering one record at a time in the SMS Administrator console.
   - Importing a product compliance data file into the SMS site database. You can create a product compliance data file as a tab-separated text file, which includes multiple product compliance records.

   For more information about these methods, see the "Customizing Product Compliance Data" section later in this chapter.
5. Ensure that the site is collecting software inventory or software metering data.
6. Run queries and reports to detect compliance issues.

When the analysis is complete, you will have a list of product compliance issues that you need to resolve.

Compliance Solutions

If, by using the data analysis results, you determine that there are software compliance issues in your organization, you can typically resolve them by applying software patches or by upgrading the noncompliant software. You can use the SMS software distribution feature to apply these solutions. In some cases, you might need to uninstall software that cannot be brought into compliance.

► To resolve product compliance issues by using software distribution

1. Create a collection of the clients that require an update by using the compliance query that you used to detect compliance issues.
2. Gather the updates that will resolve the compliance issues. Many software vendors place product updates on the Web. You can download these updates and distribute them to SMS clients.
3. Use software distribution to distribute the updates to the collection that you created.

After identifying and resolving compliance issues in your organization, it is important to ensure that users migrate to compliant applications instead of continuing to use noncompliant software. You can use software metering to monitor the use of software applications that you know are noncompliant. For more information about using software metering, see Chapter 8, "Software Metering."

After resolving the compliance issues of products that are used by SMS clients, rerun the queries and reports that you created to ensure that all issues are resolved for all clients.

For more information, see the following chapters:

- For information about collections and queries, see Chapter 4, "Managing Collections and Queries."
- For information about software distribution, see Chapter 5, "Distributing Software."
- For information about creating reports, see Chapter 11, "Creating Reports."

Viewing Product Compliance Data

When you select **Product Compliance** in the SMS Administrator console, product compliance data appears in the details pane. In this view, you can browse the product compliance records that are stored in the SMS site database.

► To view product compliance data in the SMS Administrator console

1. In the SMS Administrator console, navigate to **Product Compliance**.

```
Systems Management Server
 └ Site Database
  └ Product Compliance
```

2. Click **Product Compliance**. Product compliance records are displayed in the details pane.

The product name, displayed in the details pane, is the combination of the product's name, version, and revision numbers. When viewing product compliance data in the SMS Administrator console, you can right-click an item in the details pane, and then select **All Tasks**. If applicable, you can select **Go to Web Page** to link to the appropriate Web page.

Customizing Product Compliance Data

To analyze product compliance, the SMS site database must contain product compliance data. Product compliance data consists of records, each of which represents a product compliance guideline that needs to be included in the compliance analysis. Initially, the SMS site database does not contain product compliance data, but you can generate and add product compliance records to the SMS site database.

You can add or modify product compliance records to the SMS site database either manually or automatically. You can only delete product compliance records manually. With the manual method, you use dialog boxes in the SMS Administrator console to update product compliance data. This method is useful if you need to customize a small number of records. The automatic method requires a product compliance data file and is more efficient if you need to customize a large number of records. Customizing the product compliance data by using both methods is discussed later in this section.

When adding a new product compliance record, you can also extend the list of compliance types and compliance levels as follows:

- Add a new compliance type with a new compliance level for that type
- Add a new compliance level for an existing compliance type

If a new record contains a new compliance type or level, the new item is appended to the list of compliance types or compliance levels as appropriate. You can then view the new type or level in the **Compliance** area in the **Product Compliance Properties** dialog box. The new compliance type is listed in the **Type** list. The new compliance level is listed in the **Level** list when you select the type that is associated with that level**.**

Important

When adding or updating product compliance records, use consistent labels for compliance type and compliance level. This ensures that queries and reports yield the expected results.

To perform any operation described in this section, you need to navigate to **Product Compliance** in the SMS Administrator console, as follows:

```
Systems Management Server
 └ Site Database
  └ Product Compliance
```

Customizing Product Compliance Data Manually

The following procedures describe how to manually customize individual product compliance records.

To add a product compliance record

1. In the SMS Administrator console, right-click **Product Compliance**, select **New**, and then select **Product Compliance**.
2. In the **Product Compliance Properties** dialog box, enter the details for the product compliance record by performing one of the following steps:
   - Extract information from the product's .exe header file. In the **Product Compliance Properties** dialog box, select **Browse**. Navigate to a directory that contains the .exe file for the product that you want to add, and then select it. Your site server must have access to the .exe file of the product that you want to add.
   - Extract information from software inventory data. In the **Product Compliance Properties** dialog box, click the down arrow to the right of **Product name**. This list is populated with software inventory products from the SMS site database. Select the product that you want to add from this list.
   - Click the **General** tab, and then type the necessary information.
3. Select **Set,** and then enter name, version, and revision information to create a display name that identifies the product.
4. In the **Compliance** area, enter the type and level.
5. Click the **Information** tab, and then enter any additional information for the new product.

Note

Typing the information is not a recommended method, because if you do not enter the data exactly as it reads in the .exe header, your query results might not be accurate. Entering data by using an .exe file is a more reliable method.

To modify a product compliance record

1. In the SMS Administrator console, select **Product Compliance**.
2. In the details pane, right-click the product that you want to modify, and then select **Properties**.
3. In the **Product Compliance Properties** dialog box, modify the product information.

To delete a single product compliance record

1. In the SMS Administrator console, select **Product Compliance**.
2. In the details pane, right-click the item that you want to delete, and then click **Delete**.

To filter and then delete multiple product compliance records

1. In the SMS Administrator console, right-click **Product Compliance,** and then click **Delete Special**.
2. In the **Delete Special** dialog box, select the appropriate items from the lists to filter the records that you want to delete, and then click **OK**.

Customizing Product Compliance Data Automatically

By using a product compliance data file, you can automatically add or modify multiple product compliance records simultaneously. A product compliance data file contains information about new product compliance records, and updated information about existing records. The following sections provide information about using a product compliance data file.

Structure of a Product Compliance Data File

The product compliance data file is a tab-delimited ASCII text file that typically contains multiple lines. Each line represents a single product record. A single tab separates columns, and there is a character return at the end of each line. Table 12.1 describes the columns of each line in the file.

Note

You must include separating tabs even for an empty column.

Table 12.1 Product Compliance Data File Structure

WBEM property name	Display name	Type and length	Required?	Description
ProdName	Name	Text 35	No	The product name display name.
ProdVer	Version	Text 30	No	The product version display name.
ProdRev	Revision	Text 30	No	A display name for the specific revision within the product version.
ProdCompany	Company	Text 35	No	A display name of the company that produces the product.
ProdLang	Language	Text 35	No	A display name of the product's language.
ProdPlatform	Platform	Text 35	No	A display name of the product's platform.
Source	Data Source	Text 50	No	Identifies where the information for the product was obtained.
ResProdName	ProductName	Text 100	Yes	The product name exactly as it appears in the .exe header file.
ResProdVer	ProductVersion	Text 50	Yes	The product version exactly as it appears in the .exe header file.

(continued)

Table 12.1 Product Compliance Data File Structure *(continued)*

WBEM property name	Display name	Type and length	Required?	Description
ResProdLangID	ProductLanguageID	Numeric	Yes	The language ID for the product.
FileName	File Name	Text 255	Yes	The complete product file name.
FileSize	File Size	Integer 4	Yes	The size of the file in bytes.
Type	Compliance Type	Text 20	Yes	The compliance type.
Category	Compliance Level	Text 30	Yes	The compliance level within the specified compliance type.
URL	URL	Text 255	No	The URL path where additional information about this product's compliance might be found.
Comment	Comment	Text 255	No	Additional information about the product.

The first six fields listed in Table 12.1 are display names. You can use these fields to assign an easily identifiable name to each product. Because manufacturers do not necessarily have standards for the fields that appear in the header files of their products, it might be difficult to recognize the exact product by the header name. You can add display names to help you recognize the products that are listed in your database.

The **ResProdName**, **ResProdVer**, and **ResProdLangID** fields are primary keys that function as a group (Group1). The **FileName** and **FileSize** fields are primary keys that also function as a group (Group2). These key groups are used to compare records in the SMS site database when importing a data file.

When the product is displayed in the SMS Administrator console, the **ProdName**, **ProVer**, and **ProRev** fields are combined to make the complete product name.

Using a Product Compliance Data File

The following operations involve the use of a product compliance data file:

- Importing a product compliance data file that contains information about products that you want to add or modify. SMS processes the file and customizes the product compliance data in the SMS site database by adding or modifying product compliance records.
- Exporting product compliance records from the SMS site database to a product compliance data file. You can then import the file to the same site or to other SMS sites in the hierarchy.

When importing a product compliance data file, SMS compares each line of the imported file against each product compliance record in the SMS site database. The comparison is based on the Group1 and Group 2 primary key groups. Depending on the comparison results, product compliance records are modified or added.

A line in the data file matches a database record if, for a given source and compliance type, all of the primary key items match. The **ResProdLangID** field is ignored when the record match occurs. As a result, if a product is listed as being compliant in all languages, the compliance data is applicable to all language versions of that particular product.

To successfully import a product compliance data file, you must construct the file exactly as described in Table 12.1. Also, for an entry to be imported into the database, it must have information for all the items in Group1, Group2, or both. If one item is missing in a group, but the other group is complete, the entry is imported.

During an import, SMS treats blank fields as follows:

- If the **ProdLang** field is blank, SMS attempts to use the HDR-Prod ID to map to the appropriate language; otherwise the field is set to "unknown" in the record.
- If the **ProdPlatform** field is blank, it is set to "unknown" in the record.
- If the **Source** field is blank, it is set to the domain name/user name of the person who is currently logged on.

► To import a product compliance data file

1. In the SMS Administrator console, right-click **Product Compliance**, select **All Tasks**, and then select **Import Product Compliance Data.**
2. In the **Import Product Compliance Data** dialog box, enter the path for the product compliance data file that you want to import, and then click **OK**.

   SMS compares each line of the imported data file against each product compliance record in the SMS site database. SMS then determines whether to add a new product compliance record or to modify an existing record, as follows:

   - Any line in the product compliance data file that does not match an existing product compliance record in the SMS site database is appended to the SMS site database as a new product compliance record.
   - Any line in the product compliance data file that matches an existing product compliance record in the SMS site database replaces that record in the SMS site database.

After customizing a compliance database, you might want to share the new information with other SMS servers and sites. You can export the product compliance data from the SMS site database to a file that can be later imported into other SMS sites.

► To export product compliance records from the SMS site database to a product compliance data file

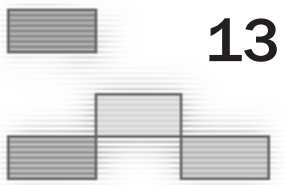
1. In the SMS Administrator console, right-click **Product Compliance**, select **All Tasks**, and then select **Export Product Compliance Data**.
2. In the **Export Product Compliance Data** dialog box, use **Export from data source** and **Export compliance type** to filter the data that is exported.

3. Enter a file name and path for the export file, and then click **OK**.

To avoid problems with extended characters, the file is exported in Unicode format. You can use a text editor, such as Notepad, to convert this file to ASCII format, if necessary.

CHAPTER 13

Maintaining and Monitoring SMS Systems



Microsoft® Systems Management Server (SMS) 2003 sites require regular maintenance to provide services effectively and continuously. Regular maintenance ensures that the hardware, software, and the site database in your sites function properly and efficiently. Use this information to develop an effective maintenance plan for your organization.

In This Chapter

- Maintenance and Monitoring Overview
- Performance Monitor Counters
- Maintenance Tasks
- Daily Tasks
- Weekly Tasks
- Periodic Tasks
- Event-Driven Maintenance Tasks
- Maintenance Throughout the Hierarchy
- Maintenance Operations

Maintenance and Monitoring Overview

After installing and setting up your SMS hierarchy, you should develop a maintenance plan. The maintenance plan should include the maintenance tasks, and site monitoring tasks that are described in this chapter.

Maintenance and Monitoring Plan

After you develop a maintenance plan for all sites, document the details of that plan so that it is easy to review and update. Having a maintenance plan document also simplifies the monitoring of maintenance throughout the hierarchy. Documenting the plan is especially important in large hierarchies where there can be many SMS administrators. You can provide the plan document to the SMS administrators that are responsible for site maintenance to ensure that sites are maintained as planned.

For each site, you should include in your plan:

- Which maintenance tasks to perform.
- How to perform each maintenance task.
- How often to perform each maintenance task.
- Who should perform each maintenance task
- How and where records about performing the maintenance tasks will be kept. Those records should include information about who performed each maintenance task, and when.

A maintenance plan also includes tasks to monitor the site activity. The purpose of monitoring the site activity is to ensure that the administrative tasks are properly performed. Monitoring is usually done by examining status messages and log files.

Maintenance and Monitoring Resources

There are various resources that you can leverage in various ways for site maintenance. Those resources are provided by SMS, the operating system, or as an application. Ensure that the maintenance plan references those resources where appropriate.

Resources for site maintenance and monitoring include:

- SMS maintenance tasks.
- SMS status system.
- SMS log files.
- SMS reports.
- SMS recovery and repair tools.
- Network diagnostic tools.

- Windows event log.
- System Monitor.
- SMS 2003 Performance Monitor counters.
- SMS Management Pack for Microsoft Operations Manager.

Maintenance tasks

To maintain SMS systems, you need to use SMS maintenance tasks. SMS provides several predefined maintenance tasks, and also allows you to create custom maintenance tasks based on SQL commands. You can schedule to run maintenance tasks automatically on a regular basis.

SMS status system

The SMS status system is an important component that helps monitor the activity at the site. The status system has many optional settings that you can configure to ensure that it is effective and useful. For example, you can configure the status system to write messages to Windows event log, or to launch a program when a certain status message is received. You can configure which messages are stored in the SMS site database and how messages are summarized. You can also decide which status messages are forwarded to parent sites and which are not.

Note

If the status system of a site is configured to not send status messages to its parent site, then you cannot use the parent site to view status messages for its child site.

When viewing status messages summaries at parent sites, it is important to remember that the information displayed in the Administrator console might be slightly out of date. You must take into account the time needed for status message summaries to be replicated up from child sites.

For more information about configuring the status system, see Chapter 14, "Using the SMS Status System."

SMS log files

Sometimes, the SMS status system might not have sufficient details when trying to evaluate the state of a component, or when troubleshooting a problem. In this case, you can review SMS log files, which provide additional details about component's activity and state. If necessary, enable logging for a component, and then examine the log files that are generated.

Because SMS depends on Microsoft SQL Server™, it is important to monitor and maintain SQL Server itself. You can use SQL Server error log files to monitor the health of SQL Server.

SMS reports

SMS provides many predefined reports that can help you monitor the status and the activity of site systems, site servers, and clients in the site. You can run predefined reports to display information such as installation status of clients, advertisements status, and hardware data for a specific client. You can also use reporting to generate custom reports as required for specific administration tasks. For more information about the SMS 2003 reporting feature, see Chapter 11, "Creating Reports."

SMS recovery and repair tools

SMS provides several tools that are used primarily during a site recovery operation, but some of them are also useful for site maintenance. The following tools can assist you in site maintenance and repair:

- SMS Site Repair Wizard—automates some Recovery Expert tasks. Using the SMS Site Repair Wizard eliminates user errors that might occur when performing complex tasks.
- ACL Reset tool — Resets access control lists.
- Hierarchy Maintenance tool (PreInst)
  Passes commands to the SMS Hierarchy Manager while the SMS Hierarchy Manager is running, to perform various site repair tasks.

For more information about using recovery and repair tools, see the "Recovery and Repair Tools" section in Chapter 15, "Backup and Recovery."

Network diagnostic tools

The Network Monitor, Monitor Control, and Network Trace tools help you monitor your network and diagnose network related problems. Network Monitor helps you to capture and analyze network frames to diagnose network problems and to identify optimization opportunities. The Monitor Control tool helps you monitor your network in real time and detect problems. By using the Network Trace tool, you can graphically display SMS network site views so you can maintain up-to-date diagrams of sites layout. For more information about network monitoring and maintenance, see Chapter 10, "Maintaining and Monitoring the Network."

Windows event log

Detailed information about hardware, software and security related events on workstations, and servers, is recorded to Windows event logs. Viewing those events helps to monitor the health of systems in the hierarchy, and to diagnose problems. You can configure the SMS status system to forward SMS status messages to Windows event log.

Note

When SMS is configured to write status messages to Windows event logs, SMS error status messages are written as Information events, not Error events.

System Monitor

You can use the Windows System Monitor tool to monitor the performance of SMS site servers, component servers, and site systems. You can configure System Monitor to send messages, e-mail, or other notifications at specified events.

SMS 2003 provides many predefined performance monitor objects with counters that you can use to monitor SMS systems performance.

SMS 2003 Performance Monitor counters

SMS provides many predefined performance monitor objects with counters that provide information about the performance and health of the various SMS components.

SMS Management Pack for Microsoft Operations Manager

Microsoft Operations Manager (MOM) can help you monitor all systems in one console. MOM provides many features for an easier, and more effective system monitoring. For more information about MOM, see http://www.microsoft.com/mom/default.asp. To download the SMS Management Pack, see http://www.microsoft.com/mom/downloads/basepack/default.asp.

Performance Monitor Counters

SMS provides a wide range of predefined performance monitor counters. You can use performance counters to monitor how SMS performance and usage impact affects system resources such as I/O and CPU. These counters can be very helpful for SMS maintenance, for identifying bottlenecks, tuning up SMS systems, and troubleshooting. Performance counters can also help you gather information about growth patterns, which can then be used to plan future hardware growth in your organization.

You can access SMS performance counters by running System Monitor (Perfmon.exe) on an SMS site server (the SMS_EXECUTIVE service must be running on that site server).

On site servers, SMS provides counters for performance monitor objects such as SMS Discovery Data Manager, SMS Executive Thread States, and SMS Inventory Data Loader. On management points, SMS provides counters for performance monitor objects, which monitor items such as the Hardware Inventory Manager, Status Manager, and policies.

Using SMS Performance Monitor Counters

You can define the circumstances that indicate that the system is not performing as usual by developing performance baselines and configuring System Monitor with corresponding thresholds. When these thresholds are met, System Monitor signals an alert. You can choose the form of the alert that is most convenient to you, and other tasks that will be automatically performed. You can choose to receive an alert email, or you can choose to run a specified application. If MOM is used in your organization, then you can configure MOM to display alerts in the MOM console.

You can configure and use System Monitor on each system, or dedicate a remote server for that purpose. Because System Monitor itself consumes system resources, using it remotely reduces its effect on the system performance, which provides more accurate monitoring.

Maintenance Tasks

A site maintenance plan consists of various maintenance tasks:

Predefined maintenance tasks SMS provides predefined maintenance tasks that you can schedule to run automatically on a regular basis.

Custom maintenance tasks You can create custom maintenance tasks by using SQL commands that you can schedule to run automatically on a regular basis.

Manual tasks Some maintenance tasks cannot be automated. You must perform these tasks manually.

This section describes the tasks that you can schedule to run automatically; you can schedule these tasks to run daily, monthly, or periodically according to the maintenance plan:

- Predefined site maintenance tasks
- Custom maintenance tasks

Predefined Site Maintenance Tasks

As your SMS system accomplishes the tasks that you schedule and configure, SMS components continually add data to the site's database. As the amount of data grows, database performance declines.

Predefined maintenance tasks target the SMS site database and help to maintain its performance. Schedule these tasks to continually remove orphaned and out-of-date data, which are of no use to you or to the site. Reducing the size of the database by removing unnecessary data improves the performance and the integrity of the database, increasing the efficiency of the site.

► To run a predefined maintenance task

1. In the SMS Administrator console, navigate to **Tasks**:

```
Systems Management Server
 └ Site Database
  └ Site Hierarchy
   └ <site code> - <site name>
    └ Site Settings
     └ Site Maintenance
      └ Tasks
```

2. In the details pane, right-click the task that you want to run and click **Properties**.
3. In the task **Properties** dialog box, enable and configure the task. To minimize interference with the site operation, set the time period to off-peak hours of the site. The time period is the time interval in which the task can run. It is defined by the **Start after** and **Latest start time** specified in the task properties dialog box.
4. Click **OK** to save your settings. The task will now run according to its schedule.

If a task fails to run at first attempt, the SMS_SQL_Monitor service attempts to rerun the task until either the task runs successfully, or until the time period in which the task can run has passed. Predefined maintenance tasks log their activity to the SMS_SQL_Monitor log file (SMSdbmon.log).

The following predefined maintenance tasks are described in this section:

- Backup SMS Site Server
- Rebuild Indexes

- Monitor Keys
- Delete Aged Inventory History
- Delete Aged Status Messages
- Delete Aged Discovery Data
- Delete Aged Collected Files
- Delete Aged Software Metering Data
- Delete Aged Software Metering Summary Data
- Summarize Software Metering File Usage Data
- Summarize Software Metering Monthly Usage Data
- Clear Install Flag

Backup SMS Site Server

This task backs up an SMS site, including the SMS site database, SMS files, registry keys, and system configuration information. Backing up your site is an essential step when preparing the site for recovery in case your site fails. When recovering a failed site, having a comprehensive site backup greatly helps to restore the site to its original state.

Use this task to back up your site on a regular basis. For more information about backup and recovery and how to use this task, see Chapter 15, "Backup and Recovery."

This task is disabled by default.

Rebuild Indexes

This task reindexes all of the site's database indexes. An index is a database structure associated with a table that speeds up data retrieval. Because of this, searching indexed columns is much faster than searching non-indexed columns.

SMS uses many indexes to maximize the SMS site database performance. However, the SMS site database indexes are frequently updated to remain synchronized with the data, which is constantly changing. Frequent updates to indexes reduce their efficiency. This reduces the SMS site database performance and efficiency.

Use this task to rebuild all indexes in the SMS site database, which increases the efficiency of the SMS site database indexes and increases site performance.

Note

In larger sites, administrators might want to have more control over the reindexing process. In this case, instead of running this task, administrators can set up an equivalent Database Maintenance Plan in the SQL Server Enterprise Manager.

This task is enabled by default, and scheduled to run every Sunday, between midnight and 5:00 A.M.

Monitor Keys

This task monitors primary keys and handles situations in which internal counters (that are used for SMS object IDs) roll over. A primary key is a column or combination of columns that uniquely distinguishes one row from any other row in a table.

Use this task to maintain the integrity of primary keys that are used in the SMS site database.

This task is enabled by default, and scheduled to run every Sunday, between midnight and 5:00 A.M.

Delete Aged Inventory History

If the hardware inventory feature is enabled at the site or at any of its child sites, then SMS is collecting hardware inventory data from clients on a regular basis, and storing that data in the site's database.

Use this task to delete from the SMS site database all hardware inventory history data older than the number of days specified in the task properties dialog box.

This task is enabled by default, and scheduled to run every Saturday, between midnight and 5:00 A.M, and to delete data that is older than ninety days.

Delete Aged Status Messages

This task deletes aged status messages. SMS generates status messages for all its activities at a fairly rapid rate. The number of status messages that are stored in the site's database, and how long they are kept in the site's database, depends on how you configured the site status system. The Delete Aged Status Messages task checks status filter rules to determine which messages need to be deleted. A large number of status messages impacts the site's database performance and uses a significant amount of disk space.

Use this task to delete aged status messages that are no longer needed from the SMS site database and from the site's hard disk.

Note

Status messages are essential when monitoring SMS sites, especially when diagnosing problems at the site. Do not delete warning or error status messages until you have reviewed them, and you are sure that they are no longer needed.

This task is enabled by default and scheduled to run every day, between midnight and 5:00 A.M. Every time the task runs, it checks the settings in the **Actions** tab of status filter rules to determine which messages have expired. By default, status messages are kept for seven days.

For more information about configuring the status system, see Chapter 14, "Using the SMS Status System."

Delete Aged Discovery Data

This task deletes aged discovery data. If client discovery methods are enabled at the site or at any of its child sites, then SMS is collecting discovery data from clients and is storing that data in the site's database.

Use this task to delete from the SMS site database all data associated with resources that are no longer considered part of the SMS hierarchy. This task identifies resources that have not reported discovery data in more than the number of days specified in the task properties dialog box. The task deletes from the SMS site database all data (including hardware inventory, software inventory, and software metering data) referencing that resource, deleting the resource from the site.

This task is enabled by default and scheduled to run every Saturday between midnight and 5:00 A.M., and to delete data older than 90 days.

Delete Aged Collected Files

This task deletes aged collected files and orphaned software inventory records. If software inventory is enabled at the site or at any of its child sites, then SMS is collecting software inventory data and possibly files (if configured to do so) from clients on a regular basis. SMS stores software inventory data and information about the collected files in the site's database, and stores the collected files on the site server's hard disk (by default, up to five instances of each collected file).

When files and products are no longer inventoried, their respective records remain in the SMS site database, but they are orphaned. Software inventory records no longer reference them.

Use this task to remove orphaned software inventory records, and to delete collected files that are older than the number of days specified in the task properties dialog box. The task deletes the aged collected files from the site server disk, and removes from the SMS site database references to these files.

This task is enabled by default and scheduled to run every Saturday between midnight and 5:00 A.M. and to delete data older than 90 days.

Note

Any aged collected files or orphaned software inventory records that are referenced by software metering data are not removed.

Delete Aged Software Metering Data

If the software metering feature is enabled in your site, then SMS is metering software on client computers, and is storing large amounts of software metering data in the site's database. Use this task to remove aged software metering data, and to conserve space in the SMS site database.

For more information about software metering maintenance tasks, see Chapter 8, "Software Metering."

Delete Aged Software Metering Summary Data

If the software metering feature is enabled in your site, then SMS is metering software on client computers, and is storing large amounts of software metering data in the site database. Use this task to remove aged software metering data, and to conserve space in the SMS site database.

Summarize Software Metering File Usage Data

If the software metering feature is enabled in your site, then SMS is metering software on client computers, and is storing large amounts of software metering data in the site's database. Use this task and other software metering maintenance tasks to summarize software metering data, and to conserve space in the SMS site database.

Summarize Software Metering Monthly Usage Data

If the software metering feature is enabled in your site, then SMS is metering software on client computers, and is storing large amounts of software metering data in the site's database. Use this task and other software metering maintenance tasks to summarize software metering data, and to conserve space in the SMS site database.

Clear Install Flag

When clients are installed, they are flagged by SMS with an **installed** status. Clients maintain their installed status even after they are uninstalled, thus preventing them from being reinstalled by the Client Push Installation method.

Use the Clear Install Flag task to flag uninstalled clients as uninstalled, thus allowing them to be reinstalled by the Client Push Installation method. If the Client Push Installation method is the only installation method enabled in the site, then you must enable this task to allow reinstallation of clients.

The task examines all clients in the database, and for each client, it checks whether the following two conditions are met:

- The client has not been recently discovered by the Heartbeat Discovery method, and not by the Windows NT Logon Discovery (SMS 2.0).
- The client is flagged as **installed** (such as when a client was previously installed, and then uninstalled.)

The task flags every client that meets both conditions with an uninstalled status. This allows the Client Push Installation method to reinstall these clients.

By default, this task is disabled. When enabling this task, set the rediscovery period to be longer then the Heartbeat Discovery interval to ensure that the task switches the client status only for clients that have been uninstalled.

Important

If heartbeat discovery is disabled, running this task is useless. It cannot identify clients with the inappropriate installation status. If you disable heartbeat discovery, then disable this task.

Custom Maintenance Tasks

You can create custom maintenance tasks by creating tasks that run valid SQL commands directly against the SMS site database. Any command that you can run in SQL Query Analyzer is a valid SQL command.

Custom SQL commands are very powerful, because they directly manipulate the SMS site database. Improper SQL commands can corrupt the SMS site database. You should secure the SMS site database by limiting the **Manage SQL Commands** SMS object security right to the appropriate SMS administrators.

► To create a custom maintenance task

1. In the SMS Administrator console, navigate to **SQL Commands**.

```
Systems Management Server
 └ Site Database
  └ Site Hierarchy
   └ <site code>- <site name>
    └ Site Settings
     └ Site Maintenance
      └ SQL Commands
```

2. Right-click **SQL Commands**, select **New**, and then select **SQL Command**.
3. In the **SQL Command Properties** dialog box, enter a name and configure the task:
   - Specify a SQL command. You can specify either a single line (255 characters) SQL command, or the name of a stored procedure that contains multiple SQL commands.
   - Specify a log file name if you want to review the results after the SQL command runs.
   - Specify a schedule for the SQL command.

Note

Before you schedule a custom maintenance task, ensure that the SQL command is valid by testing it in SQL Query Analyzer.

You can use the following SQL commands to create helpful custom maintenance tasks:

- The SQL DBCC (Database Consistency Check) command checks the integrity of the SMS site database. Schedule this task to run on a regular basis to detect database integrity problems early. Also, schedule this task to run before and after every site backup cycle.
- The SQL **xp_sqlmaint** command runs database maintenance tasks.
- The SQL **sp_who** command determines the number of SQL Server connections currently in use by SMS, or by any other process.
- The SQL **sp_spaceused** command displays the number of rows, disk space reserved, and disk space used by a table in the current database, or displays the disk space reserved and used by the entire database.
- The SQL **sp_monitor** command displays SQL Server activities and statistics.

Daily Tasks

Daily tasks include maintenance and monitoring tasks that you need to perform daily. Adjust the frequency of these tasks to fit the capacity of the site and the needs of your organization. Depending on the results of the monitoring tasks, you might need to perform additional tasks to handle problems in the site.

Daily Site Maintenance Tasks

To best maintain your system, perform the maintenance tasks in this section on a daily basis. You can automate some tasks by scheduling predefined or custom maintenance tasks to run on a daily basis.

Daily Automated Tasks

The following predefined maintenance tasks should be scheduled to run on a daily basis:

- Delete Aged Status Messages
- Backup SMS Site Server

Daily Site Monitoring Tasks

To best maintain your system, perform the following monitoring tasks on a daily basis. If there is any indication of a problem, isolate and repair the problem to ensure that the site remains healthy.

Daily site monitoring tasks include:

- Checking SMS site database status.
- Checking site server status.
- Checking site systems status.
- Checking client status.
- Checking the operating system event log.
- Checking the SQL Server error log.
- Checking system performance.
- Checking SMS system folders.

Check SMS Site Database Status

Use the SQL Server DBCC command to check the health of the SMS site database. Use any other tools available to test the health of the SMS site database.

Check Site Server Status

View site status summary information in the SMS Administrator console, or create reports that summarize the server activity and status (such as the Clients that Received a Specific Advertised Program report). If necessary, check status messages of individual components. For further details, in case status messages indicate that a problem exists, view the relevant log files. Isolate and fix conditions that generate errors or warnings. If appropriate, reconfigure the status system so that only relevant and helpful messages are recorded.

Check the status of items such as:

- Site components and services. Check if any site server component or service is experiencing any problems.
- Packages and advertisements. Check the status of packages and advertisements in your site. Check package and advertisement status messages to ensure that package source files reach distribution points, and that advertised programs reach clients. Check status messages that are returned from clients to see whether the clients run programs successfully or not.
- Site-to-Site Communication. Check communication between the site and its parent and child sites (if they exist). Check status messages and, if necessary, check log files of the Replication Manager, Scheduler, and Senders on the site to determine whether the site is having communication problems.

Check Site Systems Status

Check the state of site systems throughout the SMS hierarchy. Use the status system and, if necessary, use log files to determine if site systems are having problems, such as:

- Low level of available disk space.
- SMS components that cannot connect with a site system.

Check Clients Status

Check the state of clients in the hierarchy. Run queries on status messages to detect any problems that clients might be having, such as:

- Client components are experiencing problems.
- Clients are failing to install.
- Clients are not reporting software inventory or hardware inventory.
- Clients that are not reporting heartbeat discovery data regularly (or for the past *x* days).
- Client count unexpectedly increasing or decreasing at a fast rate.

You can monitor a client's status only if it creates status messages, and these status messages reach the site server. However, if the client, the CAP, or the management point is experiencing problems that prevent status messages from reaching the site server, you will not be aware of any problems. To detect clients from which you are missing status messages, you need to run a query that returns all clients that have not reported a status message within the last <*n*> days. In this query, <*n*> is the length of time you would expect to receive a status message from that client (taking into account the frequency of hardware or software inventory and the regular time it takes for status messages to reach the site server.)

Check the Operating System Event Log

On key servers, check the application, system, and security system event logs. You can access those through the **Event Viewer** administrative tool on Microsoft Windows® 2000 Server. Look for messages that indicate error conditions or developing problems. Isolate and repair the conditions that generate error or warning messages.

When installing an SMS site server, its default configuration is to write status messages to the event log. This helps you identify any developing problems with SMS.

Note

When SMS is configured to write status messages to Windows event logs, SMS error status messages are written as Information events, not Error events.

Save instances of the most recent event log files for future comparison. When you can compare current log files with previous log files, it is easier to detect problems that are developing. After saving the log files, you can clear them from the event log so it is easier to detect new problems.

Check the SQL Server Error Log

Check the SQL Server error log in SQL Enterprise Manager. Look for messages that indicate error conditions. Isolate and repair the conditions that generate error or warning messages.

Check System Performance

To check whether the site server and component servers have sufficient resources and that SMS site services are running optimally, you must monitor site server and component server performance. Use performance-monitoring tools such as **System Monitor** in the Performance console on Windows 2000 Server. Check the status of critical components on the site server, on the computer running SQL Server, and other SMS site systems.

SMS installs many performance monitor counters, but you can add, remove and configure counters as needed. You can also use the SQL Server performance counters.

Save performance log files for future comparison. It is easier to detect performance trends, and to identify potential bottlenecks, when comparing current performance log file to previous performance log files.

For more information about SMS performance counters, see the "Performance Monitor Counters" section earlier in this chapter.

Check SMS System Folders

To confirm that files are being sent up the hierarchy and processed correctly, you need to check the SMS system folders.

Check for backlogs in these folders on site servers. For several folders, you can use the respective performance counters to monitor the state of those folders, shown in Table 13.1.

Table 13.1 Component Folder Locations and Contents

Component/feature	Folder	Content
Data Discovery Manager	SMS\Inboxes\DDM.box	DDR files
Client Configuration Manager	SMS\Inboxes\CCR.box SMS\Inboxes\CCRretry.box	Configuration requests for client installation
Replication Manager	SMS\Inboxes\ReplMgr.box\Outbound	Replication requests
Scheduler	SMS\Inboxes\Schedule.box\ToSend	Files waiting to be sent to another site
Inventory Processor	SMS\Inboxes\Inventry.box	Hardware inventory data files
Inventory Data Loader	SMS\Inboxes\Dataldr.box SMS\Inboxes\Dataldr.box\Process SMS\Inboxes\Dataldr.box\Process\Bad Mifs SMS\Inboxes\Dataldr.box\Process\Orphans	MIF files that are being processed MIF files waiting to be processed Bad MIF files MIF files from orphan clients
Status System	SMS\Inboxes\Statmgr.box\Queue SMS\Inboxes\Statmgr.box\Statmsgs SMS\Inboxes\Statmgr.box\Futureq SMS\Inboxes\Statmgr.box\Retry SMS\Inboxes\Statmgr.box\Outbound SMS\Inboxes\Compsumm.box SMS\Inboxes\Compsumm.box\repl	Status messages

On client access points (CAPs), check for backlogs in the folders shown in Table 13.2.

Table 13.2 CAP Folder Locations and Contents

Component/feature	Folder	Content
Status messages	CAP_<site code>\Statmsgs.box	Status message files
Hardware inventory	CAP_<site code>\Inventry.box	Hardware inventory MIF files

(continued)

Table 13.2 CAP Folder Locations and Contents *(continued)*

Component/feature	Folder	Content
Software inventory	CAP_<site code>\Sinv.box	Software inventory files
Client configuration	CAP_<site code>\CCR.box	Client configuration change requests
Client discovery	CAP_<site code>\DDR.box	Data discovery files

An unusual backlog in any of these folders means that data is not being updated in the SMS site database. If there is a backlog, you need to isolate and repair the problem.

Check Status Filter Rules

Check whether it is possible to reduce the amount of traffic generated by status messages being replicated throughout the hierarchy. If the site is currently healthy, it might be possible to add status filter rules to prevent replication of status messages, which are not necessary.

Weekly Tasks

Weekly tasks include maintenance tasks and monitoring tasks that you need to perform weekly. If necessary, adjust the frequency of these tasks to fit the capacity of the site and the needs of your organization. Depending on the results of the monitoring tasks, you might need to perform additional tasks to handle problems in the site.

Weekly Site Maintenance Tasks

To best maintain your system, perform the maintenance tasks in this section on a weekly basis. You can automate some tasks by scheduling predefined maintenance tasks or custom maintenance tasks, as appropriate, to run on a weekly basis.

Weekly site maintenance tasks are:

- Weekly automated tasks.
- Delete unnecessary files.
- Delete unnecessary SMS objects.
- Produce and distribute end-user reports.
- Run disk defragmentation tools.
- Back up application, security, and system event logs.

Weekly Automated Tasks

The following predefined maintenance tasks should be scheduled to run on a weekly basis. For more information about these tasks, see the "Predefined Site Maintenance Tasks" section earlier in this chapter.

- Rebuilding Indexes
- Monitor keys
- Delete aged inventory history
- Delete aged discovery data
- Delete aged collected files
- Delete aged software metering data
- Delete aged software metering summary data
- Summarize software metering data
- Summarize software metering periodic usage data

Delete Unnecessary Files

If Management Information Format (MIF) files or IDMIFs are used to extend hardware inventory in your site, then any MIF files that are not valid are placed in the SMS\inboxes\dataldr.box\BADMIFS folder and SMS never removes them. You must empty this folder manually. If a large number of MIFs are placed in the BADMIFS folder, it is likely that a MIF generating tool is producing the MIFs with an incorrect format. Investigate and repair the cause of the bad MIFs.

Delete Unnecessary SMS Objects

Delete objects such as collections, queries, and packages that are no longer needed at the site. Deleting unnecessary objects saves disk space, reduces intersite replications, and increases performance.

Caution

When deleting a collection, any advertisements to that collection are also deleted.

Run Disk Defragmentation Tools

Over time, disk volumes on SMS site systems become fragmented. Site operations such as distributing large software packages might significantly increase fragmentation on site servers and distribution points. As fragmentation increases, disk operations take longer, thus, the overall site performance decreases.

Run disk defragmentation tools on the SMS site server and all other site systems to maintain the performance level of disk operations.

Back Up Application, Security, and System Event Logs

Windows event logs can get full, and by default, new items will start to overwrite older items. To diagnose problems, and for other reasons, it might be necessary to refer to an older event log. It is recommended that you back up Windows event logs, and store the backups in a safe and accessible location. If necessary, increase default logs file size to accommodate larger amounts of data.

Weekly Site Monitoring Tasks

To best maintain your system, perform the following monitoring tasks on a weekly basis. If there is any indication of a problem, isolate and repair the problem, to ensure that the site remains healthy.

Weekly site monitoring tasks include:

- Checking SMS site database available space.
- Checking available disk space.

Check SMS Site Database Available Space

To find the amount of space used by database devices, run the SQL Server stored procedure **sp_spaceused** against the SMS site database. For more details about **sp_spaceused**, see the SQL Server Help. Check the tempdb device at peak usage, when several instances of the SMS Administrator console are using the database and the site is actively processing objects.

Check Available Disk Space

Check the amount of available disk space on the site server, the SMS site database server, and other SMS servers. Ensure that the amount of free disk space is sufficient for SMS and SQL Server to perform properly during regular and increased activity load.

► To use the Status System to view information about site system disk space

1. In the SMS Administrator console, navigate to **Site System Status**.

```
Systems Management Server
 └ Site Database
  └ System Status
   └ Site Status
    └ <site name>
     └ Site System Status
```

2. In the details pane, view status information of site systems such as free disk space.

Periodic Tasks

Periodic tasks include maintenance tasks and monitoring tasks that you need to perform on an infrequent basis. Depending on the results of the monitoring tasks, you might need to perform additional tasks to handle problems in the site.

To optimize and protect your SMS sites, determine the frequency to schedule these tasks, based on the capacity of the site and the needs of your organization. Then, perform them according to the schedule.

Periodic Site Maintenance Tasks

To best maintain your system, perform the following tasks periodically. Use the predefined maintenance tasks when appropriate.

Periodic site maintenance tasks include:

- Backing up account data.
- Changing accounts and passwords.
- Checking network performance.
- Reviewing the security plan.
- Reviewing the maintenance plan.
- Performing recovery tests.

Back Up Account Data

To properly recover a site server, you must have information about the accounts that SMS used before the site failed. Account data is stored in domain controllers.

Use Microsoft tools, such as the NTBackup.exe tool that comes with Windows 2000 Server, or third-party tools to back up account data as follows:

- If there are multiple domain controllers in your infrastructure that contain the SMS account database, you need to periodically back up the account database. (If Active Directory® directory service is implemented in your organization, then such a task might be included in the Active Directory maintenance plan.)
- If the account database is stored on a single domain controller, then back up the account database frequently. Depending on the frequency of changes to account data, you might need to add this task to the site's daily or weekly maintenance tasks.

- If the account data is stored on member servers, then regularly back up the whole operating system that contains the account data, using software that backs up account lists and the account database.
- Whenever there is a change to the password of the Client Push Installation account or to the site system connection accounts, you should note that change. For security reasons, SMS 2003 encrypts the Client Push Installation account and the site system connection accounts. You need to be able to retrieve these accounts' passwords so that you can re-enter them during a site recovery operation.

In between account database backups, document any changes to accounts. Write down and save any changes made to SMS accounts and share rights so that you can apply those changes again after recovering the site.

For more information about restoring account data during a site recovery, see Chapter 15, "Backup and Recovery."

Change Accounts and Passwords

To maintain the level of security in your hierarchy, you must periodically change the passwords and the accounts that SMS sites use. Report any changes to the security staff so that security administrators know that these changes are planned and authorized.

To develop an effective security maintenance plan for your SMS hierarchy, you must thoroughly understand how security is deployed in your hierarchy and make the following decisions:

- Which accounts need to be changed, and for which accounts is it sufficient to change only the password.
- How often to change passwords and accounts.
- How to change passwords and accounts (such as by running SMS site reset).
- Which accounts cannot be configured by the administrator (either the account name cannot be changed, or the password cannot be manually modified).

Note

After changing passwords or accounts that SMS sites use, you must update the backup of the account database. Follow your SMS backup plan to initiate an immediate backup of the account database.

For more information about risk assessment and maintaining security in your hierarchy, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Check Network Performance

Check the available bandwidth and error rates on the networks used by the SMS hierarchy. Use Network Monitor to capture and analyze network frames so you can diagnose network problems and look for optimization opportunities.

For more information about using Network Monitor, see Chapter 10, "Maintaining and Monitoring the Network."

Review the Security Plan

SMS evolves with time. User roles change, and people might no longer need access to some or any of the SMS functions. Although most changes in access permission should be implemented after role or staff changes, you should also periodically review the access for all users or groups to identify and delete unauthorized access permissions.

The security plan implemented for the SMS hierarchy in your organization needs to support the risk assessment of your organization. As your organization changes, policies can become ineffective.

Review security-related settings such as:

- Who has access to SQL Server and to the SMS site database.
- Who can download from SMS distribution points.
- Which accounts have permissions within SMS security.

Periodically, re-evaluate the risk assessment of your organization, and then review and update the security plan accordingly.

For more information about risk assessment and SMS security issues, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Review the Maintenance Plan

Use the maintenance plan document to review the SMS maintenance plan. SMS evolves with time, and it might be necessary to adjust the maintenance plan to accommodate growth, development, and other changes in your organization.

If there were any changes in your organization's security strategy, backup and recovery strategy, or any other strategy that affects SMS, then determine if the maintenance plan needs to be adjusted to reflect these changes.

Review maintenance tasks configuration. Check the amount of data in the site database and evaluate the usefulness of that data against the amount of space that it occupies in the database. If necessary, adjust the settings that determine the number of days that data is retained in the database.

Update the maintenance plan document to reflect any changes to the maintenance plan, and then distribute it to all SMS administrators that are using it.

Perform Recovery Tests in a Test Lab

The best way to be fully prepared for a site recovery operation is to ensure that the recovery plan is adequate and that administrators are familiar with the recovery process. After you develop a recovery plan for your site, it is recommended that you perform periodic recovery tests in a test lab.

A recovery test should follow the recovery plan developed for the production environment. Plan to perform a recovery test of the central site, and of any other systems deployed in your hierarchy. A recovery test should test all phases of recovery, including:

- Backing up a site.
- Archiving the backup snapshot.
- Simulating a site failure, such as by turning a server off.
- Recovering the failed site.
- Verifying the success of the recovery operation.

You might schedule periodic recovery tests. Company policy might require that new administrators always perform a recovery test. It is strongly recommended that you always include a recovery test when testing major changes to the hierarchy.

For example, before upgrading site server operating systems, you should probably first test the upgrade in the test lab. After completing the upgrade in the test lab, you should perform a recovery test to identify any issues or adjustments to the recovery plan associated with the operating system upgrade. This ensures that if you upgrade the servers in the production environment, you will still be able to successfully recover a failed site.

Include a recovery test in every major deployment test, such as:

- A major operating system upgrade (not service pack).
- A major change to the networking infrastructure.
- New equipment deployment or building relocation.
- An SMS major version site upgrade.

For more information about backup and recovery, and about preparing a test lab for recovery tests, see Chapter 13, “Planning for Backup and Recovery,” in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*, and Chapter 15, “Backup and Recovery,” in this book.

Periodic Site Monitoring Tasks

To best maintain your system, perform the following monitoring tasks periodically. If there is any indication of a problem, isolate and repair the problem to ensure that the site remains healthy.

Periodic site monitoring tasks include:

- Checking hardware.
- Checking site’s overall health.
- Checking the backup snapshot.

Check Hardware

Even high-quality hardware occasionally fails. Sometimes, it fails gradually, so there might be early signs. Replacing hardware before it completely fails is a key step in preventing site failure. Both Windows and SMS provide performance counters, which you can use to monitor the performance and state of the hardware used in the site.

As soon as you notice any signs of hardware-related unreliable behavior of an SMS server, replace the hardware. To properly replace server hardware, you must use the Recovery Expert. For more information about swapping the computer of SMS servers, see the "Swapping the Computer of a Site Server" section later in this chapter.

Check Site's Overall Health

It is recommended that you periodically perform a more thorough health check, as follows:

- Ensure that all SMS services are running.
- Review the Status Message System for Critical status.
- Ensure that all the latest service packs are installed.
- Ensure that the latest critical security patches are installed.
- Examine the System and Application Event logs for errors.

Note

When SMS is configured to write status messages to the system's event log, SMS error status messages are written as information events, not error events.

- Run a query to determine if discovery data is being updated correctly in the SMS site database. The query should list all installed clients in which System Resource - Agent Time is not within the heartbeat interval. It is expected that some clients might be offline, but in other cases, it might indicate a problem.
- Run a query to determine if software inventory data is being updated correctly in the SMS site database. The query should list all installed clients in which Last Software Scan - Last Inventory Collection is not within the software inventory interval. It is expected that some clients might be offline, but in other cases, it might indicate a problem.
- Run a query to determine if hardware inventory data is being updated correctly in the SMS site database. The query should list all installed clients in which Workstation Status - Last Hardware Scan is not within the hardware inventory interval. It is expected that some clients might be offline, but in other cases, it might indicate a problem.

If any of these tests fail, you need to diagnose the problem and repair it.

Check the Backup Snapshot

At the end of every site backup cycle you should check the validity of the backup snapshot. Periodically, you should perform a more thorough check to ensure that the site's backup snapshots can be successfully used for recovery.

Restore a recent backup snapshot to a disk and examine file continuity, file size, and other file properties to ensure that they do not seem corrupted. Check critical files by restoring these files to their respective applications to ensure that the application can use the restored file.

Event-driven Maintenance Tasks

There are some maintenance tasks that you cannot schedule ahead of time. These tasks are associated with certain events at the site. For example, changing the Active Directory site boundaries is an event that might require that you run an SMS maintenance task in order to adjust the site to the new configuration.

This section describes events that might affect your site, and the tasks that you need to perform when those events occur, including:

- Changing Active Directory site boundaries.
- Changing TCP/IP subnets.
- Changing SQL Server configuration.
- Changing Site Server Configuration
- Changing Hardware or Software on the Site Server
- Changing Wide Area Network infrastructure.
- Making major changes that require adjustment of the test lab.

Changing Active Directory Site Boundaries

SMS site boundaries settings can be based on Active Directory sites. In this case, if there are changes to Active Directory sites in your organization, then the SMS administrator needs to be notified about the change. The SMS administrator must adjust any SMS site boundaries that are based on Active Directory sites that have changed.

Changing TCP/IP Subnets

Whenever you make changes to TCP/IP subnets in the infrastructure of your organization, such as adding or removing a TCP/IP subnet, you need to adjust SMS site boundaries to reflect the change. This is necessary to ensure that clients are assigned correctly to SMS sites.

Clients are assigned to sites according to the site's site boundaries and roaming boundaries. Because those boundaries can be based on IP Subnets, making changes to IP Subnets in the site's site boundaries, or in the site's roaming boundaries, changes client site assignments. Changes might leave clients unassigned, causing them to uninstall their SMS client components. To ensure that clients are not uninstalled when adding TCP/IP subnets, you must add the new IP subnet as a site boundary in the site's properties before the new physical subnet is connected.

Changing SQL Server Configuration

To properly recreate database devices, and to configure SQL Server before restoring a failed site, you must have the SQL Server configuration information (such as user connections, locks, and memory) and complete device allocation information (such as device size and type).

Run SMSSQLinfo.bat from the SMS\Bin\i386 folder to back up SQL Server configuration information whenever you make changes. For more information, see the SQL Server Help.

Changing Site Server Configuration

To correctly recover a site, you must have all site server configuration data available. Whenever the following information changes, you need to update the hierarchy layout document and the site configuration document:

- Hierarchy design, including data such as parent-child relationships, primary and secondary site locations, and the computers on which SQL Server is running.
- Site configuration for each SMS site in the hierarchy, including data such as feature settings and component settings.
- Security information such as user names and passwords for client connection accounts, service accounts, and network access accounts.

Ensuring that this data is always up-to-date helps you in case a backup snapshot is not available for a recovery operation, and in the event that there is no staff familiar with the hierarchy deployment.

You can use the Network Trace tool to create and print SMS network site views to help you maintain up-to-date diagrams of site layout. For more information about using the Network Trace tool, see Chapter 10, “Maintaining and Monitoring the Network.”

Changing Hardware or Software on the Site Server

You should have a system recovery solution for SMS site servers. This allows you to recover the site server computer if it starts failing. For site servers running any of the Windows 2000 Server family, use an Emergency Repair Disk. For site servers running Microsoft Windows Server™ 2003, Standard Edition; Windows Server 2003, Enterprise Edition; or Windows Server 2003, Datacenter Edition, use the system’s Automated System Recovery feature.

Update the system recovery data every time you make significant changes to the system’s hardware or software. Changes that require an update include changes to the partition structure, changes to device drivers, and installing new applications.

Changing Wide Area Network Infrastructure

Whenever changes are made to the wide area network (WAN) infrastructure used in the hierarchy, check addresses and senders settings throughout the hierarchy, as appropriate. Ensure that they correctly reflect the new WAN infrastructure. Analyze the changes and determine if you need to add new SMS sites in order to accommodate the WAN infrastructure.

Making Major Changes That Require Adjustment of the Test Lab

If you are using a test lab, then you need to maintain it, so it continues to correctly represent your production environment. If you make changes in the production environment such as adding new client platforms, changing your network infrastructure, or making changes to feature sets, then you must make the corresponding changes in the test lab. The test lab must correctly represent the deployment hierarchy so that the results of any predeployment tests that you perform in the test lab are reliable.

For more information about the test lab see Chapter 7, "The Pre-Planning Phase," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Maintenance Throughout the Hierarchy

An SMS hierarchy can be very large and include many sites. Developing an individual maintenance plan for each site is not efficient. It is more efficient to consider the entire hierarchy when developing a maintenance plan. Developing a hierarchy-wide maintenance plan, instead of individual maintenance plans for each site, can reduce the overall maintenance workload and the overall maintenance expense.

This section describes strategies that help increase efficiency and simplify maintenance throughout the hierarchy.

Task Automation

You should automate tasks whenever possible by developing and running automation scripts. You can develop scripts that perform various maintenance and monitoring tasks. Those scripts can then notify you if any problems are encountered. You can run similar automation scripts on multiple sites.

You can automate maintenance tasks such as:

- **Monitoring performance** Monitoring performance on each site server can be inefficient. Instead, you can configure either Microsoft or third-party management tools and performance monitoring tools, to centrally monitor multiple system performance, and to send alerts, e-mails, or other notifications when certain error conditions are encountered. For example, you can implement MOM to assist with monitoring multiple SMS site servers and other systems in your hierarchy.
- **Checking for backlogs** Checking for component backlogs in SMS folders can be time consuming. Instead, you can use a third-party tool or develop a script that automatically scans all SMS folders throughout the entire hierarchy. The tool or script can search for file backlogs, files older than a set date, or files that are not valid. Configure the utility or script to send e-mail or another notification when certain error conditions are encountered.

Sharing Custom Maintenance Tasks

If you created custom maintenance tasks, then you can create similar tasks in other sites. After you have developed and tested the new custom tasks on one site, you can use a similar custom task at other sites.

Maintenance Groups

Define maintenance groups and develop a generic maintenance plan for each maintenance group. For example, you can define a secondary sites group. Develop a maintenance plan for the secondary sites in your hierarchy, and apply that plan to all secondary sites in your hierarchy. Maintenance groups can be based on geographical regions, site role, or site capacity.

Define groups as appropriate in your organization, and assign as many sites as possible to each group. That way, multiple sites can share a single maintenance plan.

Monitoring Maintenance

Develop a maintenance monitoring system to help you monitor the ongoing performance of maintenance tasks on all sites throughout the hierarchy. Especially in large hierarchies, it can be difficult to manually monitor maintenance activity at all sites.

You can use e-mail messages, status messages, or any other method that allows senior SMS administrators to ensure that all sites are getting their regular maintenance. Compare the maintenance activity to the documented maintenance plan for each site to ensure that all sites are well maintained.

Maintenance Operations

After setting up and configuring an SMS site, you might need to change site configuration to accommodate changes in your organization. Not all changes are supported. For example, you cannot change settings such as:

- The drive letter on which SMS is installed.
- The site code.
- The site name.
- The computer name of the site server.
- The parent site of a secondary site.
- Moving the SMS site database from a remote server to the site server.
- Promoting a Windows 2000 member server that is an SMS site server to be a domain controller.

To accommodate changes, it might be necessary to perform maintenance operations on the SMS hierarchy. You might want to upgrade the operating system of an SMS server, or you might need to make changes to the network infrastructure, or to relocate hardware. This section lists supported maintenance operations, and provides information about performing these operations.

Before implementing changes to the entire hierarchy, it is recommended that you first test these changes in a test lab. When testing the proposed change in the test lab, also perform a site backup and recovery test. If the proposed change requires an adjustment to the site's backup and recovery plan, then make the necessary changes.

For more information about the test lab, see Chapter 7, "The Pre-Planning Phase," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

For information about client related maintenance operations, such as replacing a Legacy Client with an Advanced Client, see Chapter 17, "Discovering Resources and Deploying Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Supported maintenance operations include:

- Attaching one site to another (creating a child site).
- Swapping the computer of a site server.
- Rebuilding the computer of a remote SMS site database server.
- Moving the SMS site database.
- Resetting the site by running SMS site reset.

Attaching One Site to Another (Creating a Child Site)

Most SMS deployments require building an SMS site hierarchy. This involves attaching one site to another, making the attached site a child site, and making the second site a parent site. After attaching one site to another, the important thing that happens is that data starts to flow from the child site to its parent site, and from the parent site to its child site.

Objects, such as software distribution related objects, collections, and queries, are replicated to the child site, and can be used at the child site. However, those replicated objects are locked at the child site. The administrator at the child site cannot modify inherited objects. For example, the administrator at the child site can use an inherited query definition to query the child site's database.

Predefined objects, such as predefined queries that are inherited from a parent site, overwrite the corresponding objects at the child site. If you modified any of those objects at a site that you plan to attach to another site, ensure that the changes are backed up, or plan to make those changes again after the attachment process is complete.

Data such as status messages and inventory is replicated from child sites to their parent site, and then from the parent site to its parent site. This ensures that ,eventually, the central site contains data from all the clients of all sites in the hierarchy.

Swapping the Computer of a Site Server

This section provides information about moving a healthy and a functioning SMS 2003 site server, in a primary or secondary site, from one computer to another. This might be necessary if the current site server computer is exhibiting signs of imminent hardware failure, or if the lease expires on a server hosting SMS roles.

Swapping the site server computer consists of the following basic steps:

1. Preparing for the hardware swap.
2. Building the new site server.
3. Restoring the SMS site data.
4. Repairing and synchronizing the data.
5. Verifying the functionality of the new site server.

A site server computer swap process is somewhat similar to a site recovery process. However, because the current site server is properly functioning, you can plan and completely control the operation. The operation is significantly simpler then a recovery operation, and no data is expected to be lost.

Because a hardware swap operation is similar to a site recovery operation, you can use the Recovery Expert to perform this operation. The Recovery Expert is designed to handle this scenario, and when you run it, it provides the appropriate tasks for a hardware swap operation.

► To swap the computer of a site server by using the Recovery Expert

1. Set up the Recovery Expert Web site if necessary. For more information about setting up and running the Recovery Expert, see Chapter 15, "Backup and Recovery."
2. Run the Recovery Expert.
   - Answer **Yes** to the question **Do you need to move a site server to different hardware?**
   - Answer **Yes** to the question **Are you recovering the site because you need to re-install the operating system?**
3. Perform the prescribed tasks in the prescribed order. Use the SMS Site Repair Wizard and other repair tools as instructed by the Recovery Expert.

For information about the Recovery Expert, and about using it to swap hardware, see Chapter 15, "Backup and Recovery."

Rebuilding the Computer of a Remote SMS Site Database Server

This section provides information about moving a healthy and a functioning remote SMS 2003 site database server from one computer to another. This might be necessary if the remote site database server is exhibiting circumstances such as the following, and you need to rebuild the computer:

- The computer is exhibiting signs of imminent hardware failure.
- The operating system is failing and it is necessary to reinstall the operating system.
- SQL Server is failing and it is necessary to reinstall SQL Server.

You can follow this procedure for any other restoration operation of a remote site database server. As long as the SMS site data on the remote database server is intact, you can treat the operation as a hardware swap operation.

► To rebuild the computer of a remote site database server

This scenario assumes that the operating system, SQL Server, and data files are each on a separate drive on the database server.

1. Stop SMS services at the site as follows: all sites sharing the SQL Server being swapped must have their SMS_SITE_COMPONENT_MANAGER service and their SMS_EXECUTIVE service stopped to prevent any process from accessing the databases.
   - Stop the SMS_SITE_COMPONENT_MANAGER and SMS_EXECUTIVE services on the site server.
   - Stop the SMS_SQL_MONITOR service on the SMS site database server.
   - Stop the Windows Management Instrumentation service on the site server, and on the SMS site database server.
   - Disable these services to prevent them from restarting. To disable a service, right-click the component name, and then click **Properties**. From the **Startup type** drop down menu on the component **Properties** page, select **Disabled**.
2. Rebuild the operating system on the failing computer, or use a different computer with an intact operating system.
3. Reinstall SQL Server. Restore the master, model, and msdb system database from the last backup to recover SQL Server permissions and definitions.
4. Recover SMS database in place without restore (see SQL Server documentation about recovering and reconnecting to existing data files in place). The data is now intact and still synced up with the site
5. Inspect databases to ensure that they can be accessed and appear to contain valid data.
6. Run SMS Setup and perform a site reset by selecting the **Modify** option.

Moving the SMS Site Database

To accommodate growth and other changes in your organization, you might want to move the SMS site database to a remote database server. Moving the SMS site database to a remote server is performed in the same manner for the following two scenarios:

- Moving a local SMS site database to a remote server
- Moving the SMS site database from one remote server to another remote server

When moving the SMS site database server, SMS moves the SMS Provider to the same server that the SMS site database is moving to

► To move SQL Server to a remote server

1. Install SQL Server on the remote server.
2. Configure the new SQL Server database security so it is similar to the old SQL Server security configuration. Configure accounts and permissions required for SMS to manage SQL Server.

3. Stop the following SMS services:

   On the site server:

   - SMS_SITE_COMPONENT_MANAGER
   - SMS_EXECUTIVE

   On the old database server:

   - SMS_SQL_MONITOR_<*SiteServerName*>

4. Back up the SMS site database from the old SQL Server database by using the SQL Server Backup Database tool.
5. Restore the SMS site database to the new SQL Server database by using the SQL Server Restore Database tool.
6. On the site server, run the SMS Setup **W**izard:
   - On the **Setup Options** page, select **Modify or reset the current installation**.
   - On the **Database Modification** page, specify the new database server name and the appropriate database name.
   - Complete the SMS Setup Wizard.
7. You can now delete the SMS site database from the old SQL Server database, or disconnect the old database server (if it is on a remote computer).

Note

While using the new SMS site database server, SMS generates error status messages as it continues to attempt to access the old SQL Server. You can safely ignore those status messages.

Resetting a Site by Running SMS Site Reset

SMS site reset stops the core SMS site services, removes them, and then reinstalls them. By using SMS site reset, you can do the following:

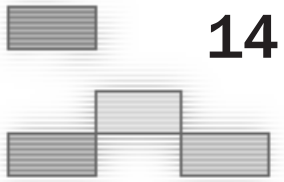
- Repair a damaged site server.
- Make account or password changes effective.
- Force the site to use a different SQL Server database name or a server running SQL Server, or a different SQL Server security mode.

If there has been a change to the accounts used by the SMS components, site reset ensures the account details used by the SMS components are correct.

The exact effect of site reset depends on how you initiate it and which options you chose. For more information about site reset, see Chapter 5, “Understanding SMS Security,” in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

CHAPTER 14

Using the SMS Status System



The Microsoft® Systems Management Server (SMS) 2003 status system monitors and reports the overall state of SMS sites and hierarchies. You use this tool to assess the health of your SMS system. You view continual site checkups that make you aware of the functional status of your site, be notified of problems before they become critical, and discover the source of problems that do occur.

Because the SMS status system has already been configured to provide you with the important data you require to troubleshoot your SMS sites, most of this chapter explains how the status system works.

In This Chapter

- Understanding Status Messages
- Interpreting System Status
- Configuring the SMS Status System
- Using the SMS Status System with the Windows Event Log

Understanding Status Messages

SMS uses status messages to report information about the health of your site. Status messages are generated from both client and server components. You can also configure SMS components to report only certain types of messages based on their severity. You can then specify the rules that determine whether they are replicated to the parent site, written to the SMS site database, or reported to the Windows Event Log. You write messages to all three places or any combination of destinations you desire.

SMS includes powerful status summarizer components that produce summaries from status messages. These summaries give you a quick, real-time view of the health of SMS sites, components, packages, and advertisements. From these summaries, you get a more detailed look at a specific problem by accessing detailed status messages about the flow of activity.

The SMS status system is designed to provide the data required to troubleshoot your site. After you become more familiar with the system, you might want to configure the status system for more site-specific details. To use the status system, you must understand status messages and their characteristics.

Status Messages Defined

A status message is similar to a Windows event or a message written to a log file. SMS components generate status messages while they carry out their tasks. If you think of SMS as a configuration management business, server components are the workers. Client components also generate status messages. Server components report many more status messages than client components for the following reasons:

- SMS servers perform significantly more work than clients.
- If something malfunctions at an SMS client, it has little effect on the rest of the organization.
- If an SMS server malfunctions, the effect on the organization might be quite large.

Similar to a Windows event or a message written to a log file, a status message is a text message generated at a particular time by a particular component. Some status messages are meaningful by themselves.

Other status messages are meaningful only within a sequence of status messages, as in the example below.

```
Beginning installation of server RED1.

Could not copy file C:\SMS\Red1.dat to \\Red1\D$\SMS\Red1.dat. The operating
system reported error 5: access is denied.

Failed to complete installation of server RED1.
```

While components carry out their tasks, they report status by creating status messages periodically and adding them to the SMS status system. They are then stored in the SMS site database. The Status Message Viewer accesses the site database to display individual status messages.

There are two broad categories of status messages.

Flow-of-activity messages Components generate flow-of-activity messages to illustrate the tasks a component is carrying out. These messages:

- Educate the administrator about the tasks the component performs.
- Facilitate complex problem debugging.
- Facilitate SMS activity auditing.
- Provide reports and summaries showing the overall health or progress of a specific SMS feature.

Error messages Components generate error messages when they encounter a problem performing a task. These messages warn you about a problem that might require your attention.

Status Message Characteristics

Status messages have several characteristics that help you identify the types of activities occurring at your sites.

Message Severity

Every status message is stamped with a severity that indicates the gravity of the information presented in the message.

Error messages Error messages are exceptional messages that occur when there are problems that require immediate administrator attention. Typically, an error message means that a component cannot recover from or work around a problem. As a result, one or more SMS features are nonfunctional until the administrator corrects the problem. Examples of the types of problems error messages indicate are:

- A server is down.
- A disk or database is full.
- Security problems have occurred.
- Critical files or data are corrupt or missing.
- An attempt to retry an operation has failed enough times that it is no longer a temporary problem.

Warning messages Warning messages are generated when potential problems occur that might require administrator attention. Typically, the component can work around these problems. Warning messages might indicate:

- A disk or database is low on storage space.
- A non-critical file, such as a Management Information Format (MIF) file or discovery data record (DDR) from a client, has been corrupted.
- An operation is taking an unusually long time to be completed.
- The administrator has configured the system in a dangerous or inappropriate way.
- An operation failed, but due to other considerations, the failure should be temporary, and the component retries the operation later.

Informational messages Informational messages are usually flow-of-activity messages that illustrate the activities of a component during normal, successful operation. Informational messages might also indicate problems if the problems are relatively unimportant. For example, if a component fails to connect to a computer by using one account, but there are three other accounts it can try, the failure is reported as an informational message. If all of the accounts fail, the component might report a warning or an error message.

Message Type

Every status message is stamped with one of three possible types.

Milestone When SMS components complete complex operations, milestone messages indicate successful or unsuccessful completion. If the operation is successful, the milestone message is informational. If the operation is unsuccessful, the milestone message is either a warning or an error message.

Detail The context of status messages representing a complex operation.

Audit Audit status messages provide an audit trail of actions that you take in the SMS Administrator console that result in objects being added, modified, or deleted. All audit messages are informational messages.

Message Date and Time

Knowing how date and time are displayed with status messages helps you to monitor and troubleshoot specific tasks that you schedule by using the SMS Administrator console. SMS stamps every status message with the system date and time that the message is reported, down to millisecond resolution. When you install SMS, the time zone in the Status Message viewer is set to the same time zone as your SMS site server.

You can configure the Status Message Viewer to display time in either **local time** (the time the event occurred in the time zone it occurred) or Coordinated Universal Time (UTC, formerly Greenwich Mean Time). You can use UTC to debug problems in an SMS hierarchy that has multiple sites in different time zones. When you use UTC, if you send a package from the New York site, you can view the messages in the Status Message Viewer to see when the package left the New York site and when it arrived in Los Angeles. It is easier to determine how events happen relative to one another when they are both reported in UTC.

Other Message Characteristics

There are other status message characteristics that help you identify the source of status messages within your SMS hierarchy.

Source Server components, client components, or the SMS provider can generate status messages.

Site code The site code that reports the status message is included with each message.

System name The system name is the name of the computer that reported the status message.

Component name The component name lists the name of the component that reported the message.

Optional properties Each status message might contain one or more optional properties. The current possible optional properties include Collection ID, Package ID, and Advertisement ID. For example, when the Available Programs Manager (a client component) reports a status message that it received an advertisement, the Available Programs Manager attaches an optional property to the status message that specifies the Advertisement ID.

Optional properties provide a consistent mechanism for different components to report status messages about particular SMS objects, such as collections, packages, and advertisements. These properties also provide a means for you to run queries about specific objects.

Miscellaneous characteristics Each status message also includes the process and thread ID that reported the message. Some status messages contain additional information, such as an error code reported by the operating system. In this case, SMS displays the exact error code reported by the operating system and the text for that error code, such as "Access is denied" (error code 5).

The Status Message Viewer

With the Status Message Viewer, you view status messages by querying the SMS site database through the SMS Provider. You can launch the Status Message Viewer from various places within the SMS Administrator console. For example, the status summaries contained in the SMS Administrator console show you the condition of your SMS site, components, packages, and advertisements. When you detect a problem at this level, you can start the Status Message Viewer to identify the problem more specifically.

To launch the Status Message Viewer, navigate to **Component Status** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ System Status
   └ Site Status
    └ site code - site name
     └ Component Status
```

Select **Component Status**. When the component list appears in the details pane, select a component, right-click, and then click **Show Messages**. Each time you launch the Status Message Viewer, you must select which messages to view based on message severity:All, Errors, Warnings, or Informational. Select one of these items. The Status Message Viewer appears and displays the status messages for the selected component on the site system that you selected.

Status messages are also based on the queries you specify in the menus leading to the viewer. The different queries you can specify depend on the where you are in **Site Status** when you launch the viewer. For more information, see the "Interpreting System Status" section later in this chapter.

Alternatively, you can launch the Status Message Viewer from **Package Status** and **Advertisement Status**. Table 14.1 describes the Status Message Viewer columns.

Table 14.1 Status Message Viewer Columns

Column name	Description
Severity	Error, warning, or informational.
Type	Milestone, detail, or audit.
Time/date	The time and date the status message was generated.
Site code	The site code where the status message was generated.
System	The computer name where the status message was generated.
Component	The name of the SMS component that generated the status message.
Message ID	The message ID of the status message.
Description	The text of the status message.

When you double-click a status message in the viewer, the **Status Message Details** dialog box appears with more details about the message, including the Process ID, Thread ID, message text, and optional message properties.

Table 14.2 summarizes the message viewer features that help you use the viewer.

Table 14.2 Status Message Viewer Features

By using this feature	You can do this
Click-Drag	Change the column widths. Add and remove columns and change their display order.
Copy/Paste	Copy status messages to your clipboard with tab- or comma- delimited columns to support paste to Excel and text editors.
Delete	Delete selected status messages from the SMS site database.

(continued)

Table 14.2 Status Message Viewer Features *(continued)*

By using this feature	You can do this
Filter on a Per Column Basis	Filter on a column basis by selecting the filter icon. Filter on specific characteristics such as message severity, message type, and site code,.
Find	Find specific status messages or all messages with specific characteristics such as message severity, message type, and site code.
Font Selection	Change the font type used to display status messages.
Multiple Viewers	Launch multiple status viewers to compare messages for troubleshooting purposes.
Multi-Selecting Rows	Select multiple rows in the Status Viewer by using the standard Shift + Select and Ctrl + Select methods.
Save and Print	Save and print status messages by using comma or tab delimiters and by using multi-select or All for options. Saving status messages to files allows you to save specific sets of status messages that might be important beyond the routine, seven-day period. By default, SMS deletes most status messages after seven days. Due to the large number of status messages generated, it is important that you configure them to be deleted frequently.
Sort on a Per-Column Basis	Sort all messages based on a specific column by clicking on a column title. The messages appear sorted based on the column you select.

Interpreting System Status

The SMS status system monitors and reports the overall health of SMS sites and site hierarchies. You use the SMS status system to view a summary of the site system, component, package, and advertisement statuses in your site. You view these summaries at different levels of detail by using status summaries.

Status summarizers produce status summaries from status messages and other data in the site database. Status summaries are not reports because they are not based on scheduled queries to the site database. Instead, they are produced in real time as the summarizers receive status messages from the other SMS components.

SMS includes four status summaries:

- Component Status
- Site System Status
- Package Status
- Advertisement Status

Each of the summaries is produced by a unique summarizer thread component, except for Package Status, which is produced by Distribution Manager.

Status Summarizer Concepts

Six important concepts relate to status summarizers and the summaries they produce. These concepts include:

- Counts versus states.
- Display intervals.
- Status indicators.
- Thresholds.
- Launching the Status Message Viewer and other tools.
- Replication of status summaries up the SMS hierarchy.

Counts and States

Each piece of data in a status summary is classified as either a count or a state. A count is a tally of events that occur over a specific period of time, such as the number of Error status messages reported by SMS Executive since the beginning of the week. A state is the last known condition of something, such as the number of free bytes available for the site database.

Each of the status summaries contains some state data. Only two of the status summaries, Component Status and Advertisement Status, contain count data. The "Monitoring and Troubleshooting with System Status" section later in this chapter tabulates and classifies the data for each summary as either a count or a state.

Display Intervals

Status summaries contain count data and tally events over a specific period of time called a display interval. Each status summary contains several different display intervals. You view count data using different display intervals to gauge when events occurred. For example, the Component Status summary counts the number of Error, Warning, and Informational status messages reported by the server components at your site. To view the counts for different periods of time, right-click **Component Status**, click **Display Interval,** and choose the period of time you would like to view. After you select a display interval, the details pane displays data based on the display interval you select.

Note

Display intervals are not available for Site System Status and Package Status Summaries because both of these summaries are based entirely on states.

Each display interval has a start time and an end time that specifies the time over which the counts are made. The start time is the time or date that you select as the display interval. The end time is always the current time. For example, if the current date and time is Wednesday at 2:32 P.M., and you select **Since 8:00:00 AM** as the display interval in **Component Status**, the summary displays counts made from Wednesday at 8:00 A.M. to Wednesday at 2:32 P.M. Or, if you select **Since Monday** as the display interval, the summary displays counts made from Monday at midnight to Wednesday at 2:32 P.M.

Both the start time and end time for a display interval are in the time zone of the site server for the site the data applies to. For example, suppose the site server for your NYC site is on Eastern Time and the site server for your site CHI is on Central Time. If you select **Since 12:00:00 AM** as the display interval in **Component Status**, the counts displayed for site NYC are since midnight Eastern Time and the counts displayed for site CHI are since midnight Central Time.

Note

To ensure the accuracy of status summary data, ensure the system dates and times on your site systems and clients are set as closely as possible to the actual date and time.

Display intervals that have earlier start times show counts that are greater than or equal to the counts shown for display intervals that have more recent start times. If the current day is Wednesday, the **Since 8:00:00 AM** display interval shows lesser or equal numbers of Errors, Warnings, and Informational status messages than the **Since Monday** display interval. If the current day is Monday, the counts are the same.

You can select different display intervals to determine when events occurred in the past. If the current time is 2:32 P.M., to determine the number of Error status messages reported by the SMS Executive between 8:00 A.M. and 12:00 P.M. on the same day, subtract the number of errors displayed when the **Since 8:00:00 AM** display interval is selected from the number displayed when **Since 12:00:00 PM** is selected.

Status Indicators

SMS uses three indicators to report the health of server components in the Component Status summary, and of site systems in the Site System Status summary.

Critical status. When Critical status is reported, there are problems that require immediate administrative attention.

Warning status. When Warning status is reported, there are potential problems that might require administrative attention.

OK status. When OK status is reported, there are no problems that currently require administrative attention.

Note

Package Status and Advertisement Status do not use status indicators.

Thresholds

A threshold is a limit you set to control when a Warning or Critical status indicator is displayed instead of an OK status indicator. For Component Status, you specify how many Error, Warning, or Informational messages a component must report before its status indicator is changed to Warning or Critical. For Site System Status, you specify how low the amount of free storage space can drop before a storage object on a site system is flagged with a Warning or Critical status indicator.

You view the thresholds that are configured for components and storage objects by right-clicking items in the details pane within **Component Status** or **Site System Status**, and selecting **Properties**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ System Status
   └ Site Status
    └ site code - site name
     └ Component Status
     └ Site System Status
```

For information that describes how to change the thresholds to meet your needs, see the "Configuring the SMS Status System" section later in this chapter.

Note

Package Status and Advertisement Status do not use thresholds.

Launching the Status Message Viewer and Other Tools

Status summaries provide high-level views of the status of your sites and hierarchy, but they might not give you enough details to solve a problem. From many places in the status summaries, you can right-click **Show Messages** to start the Status Message Viewer. In all of these places, you can launch the Status Message Viewer with a query generated in real time based on where you right-clicked in the SMS Administrator console. The query returns the subset of status messages currently in the site database that are applicable to your right-click location.

You can also launch other tools such as Resource Explorer, Network Trace, and Windows Diagnostics by right-clicking in status summaries. In all cases, the tool is launched for the specific object you select. Typically, the object is a site system.

Replication of Status Summaries Up the Site Hierarchy

To distribute processing load down the site hierarchy, each summarizer produces only a status summary of its own site. Periodically, each summarizer replicates summaries for its own site and its child sites up the site hierarchy to its peer at the parent site. The parent site summarizer integrates these summaries into its collection of summaries. Administrators at the parent site view the summaries for the parent site, and all of the sites below it in the hierarchy.

For each summarizer, you specify the replication priority (High, Medium, or Low) it uses when replicating summaries to the parent site. You can also disable the replication of summaries to the parent site. For more information, see the "Configuring the SMS Status System" section later in this chapter.

Note

Package Status summaries are always replicated up the site hierarchy at Medium priority. You cannot configure or disable the replication priority of Package Status summaries.

Status summaries replicate up the site hierarchy independent of the actual status messages that the summaries are based on. You can configure SMS to replicate only the status summaries, only the status messages, both, or neither. You might also configure the messages to replicate at different priorities. The configuration you use has a strong influence on how effectively you monitor a hierarchy of child sites from a parent site.

One possible configuration is to replicate summaries at Medium or High priority and replicate status messages at Low priority. This approach usually means the summaries are available quickly at the parent site. However, the status messages that usually comprise a large amount of data take longer to replicate. If you use this configuration and attempt to debug a problem by launching the Status Message Viewer, you might find there are no status messages in the database related to your problem. If this occurs, the messages could still be in the process of replicating up the hierarchy. If you do not want to wait for the replication to be completed, you can connect the Status Message Viewer to the child site's database (if the child site is a primary site). To accomplish this, right-click the child site's status node, and then launch Status Message Viewer.

Monitoring and Troubleshooting with System Status

The SMS status system assists you in monitoring and troubleshooting your SMS sites. You monitor the status of your SMS hierarchy using **System Status** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ System Status
   └ Status Message Queries
     └ Package Status
     └ Advertisement Status
    └ Site Status
     └ site code - site name
      └ Component Status
      └ Site System Status
```

System Status contains the four status summaries and a collection of general-purpose queries for launching the Status Message Viewer.

This section describes how you use the status system for monitoring and troubleshooting network problems with SMS. For more information, see the "Configuring the SMS Status System" section later in this chapter.

Important

Do not attempt to configure the status system until you have used it and are confident that you fully understand the ramifications of changing the various default settings.

In the following sections, it is assumed that you have navigated to **System Status** in the SMS Administrator console and that you can select a particular status summary. The next section details how you use each of the four status summaries to monitor and troubleshoot your SMS site and hierarchy.

Site Status

The Site Status item in the SMS Administrator console contains the Component Status and Site System Status summaries for the current site and all of the sites below it in the hierarchy. The Site Status console tree item has three levels: the all-sites level, the per-site level, and the status summary level. The all-sites level contains a rolled-up view of the per-site level, which contains a rolled-up view of the status summary level. The status summary level is composed of the Component Status summary and the Site System Status summary.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ System Status
    └ Site Status
     └ site code - site name
      └ Component Status
      └ Site System Status
     └ site code - site name
      └ Component Status
      └ Site System Status
     └ site code - site name
      └ Component Status
      └ Site System Status
```

The easiest way to understand the Site Status console tree is to start at the status summary level and work up to the all-sites level at the top.

Component Status

The Component Status summary is a high-level view of the health of SMS server components at a given site. The details pane includes one row for every server component running at the site. Table 14.3 describes the data reported for each component.

Table 14.3 Component Status Details Pane Columns

Column name	Data type	Description
Status	State	Health of the component: OK, Warning, or Critical.
Site system	State	Name of the site system the component is installed on. Some components might run on more than one site system.
Component	State	Name of the component.
State	State	State of the component: Installing, Started, Paused, Stopped, or Deinstalling.

(continued)

Table 14.3 Component Status Details Pane Columns *(continued)*

Column name	Data type	Description
Error	Count	Number of Error status messages reported by the component during the display interval.
Warning	Count	Number of Warning status messages reported by the component during the display interval.
Informational	Count	Number of Informational status messages reported by the component during the display interval.
Type	State	Specifies whether the component runs according to a schedule or auto-starts.
Next scheduled	State	Time and date when the component is next scheduled to start, if the component runs according to a schedule.
Last started	State	Last time and date when each component started.
Last message	State	Last time and date a status message was received from the component.

To refresh this data, right-click **Component Status** in the console tree and click **Refresh**.

The **Errors**, **Warnings**, and **Info** columns tally the numbers of Error, Warning, and Informational status messages reported by each component during the display interval. You change the display interval by right-clicking **Component Status**, selecting **Display Interval**, and then clicking the appropriate interval.

Note

The display interval for Component Status for one site does not change when you change the display interval for another site. You must specify each site's display interval individually.

You launch the Status Message Viewer by right-clicking a component and clicking **Show Messages**. The messages you view are the ones reported by that particular component during the display interval. You can launch other tools by right-clicking a component, pointing to **All Tasks**, and then selecting the tool you want to launch.

The status indicator in the **Status** column represents the health of each component. The status indicator is set based on whether the component has reported enough Error, Warning, or Informational status messages to exceed Warning or Critical thresholds. You view the thresholds for a component by right-clicking a component in the details pane and clicking **Properties**. The **Status Threshold Properties** dialog box specifies Warning and Critical thresholds for Error, Warning, and Informational status messages and the **Threshold period**. The threshold period specifies the period of time for which the thresholds are evaluated. For example, if the threshold period is **Since 8:00:00 AM** and the Informational messages threshold is set to 100 for Warning and 200 for Critical, then the component's status indicator is set to Warning whenever the component has reported between 100 and 199 Informational status messages since 8:00:00 A.M. on any given day. The component's status indicator is set to Critical whenever the component reports 200 or more Informational status messages.

Important

Do not confuse the threshold period with the display interval. Selecting a different display interval does not change the threshold period. Because the status indicators for the components are driven from the threshold period, changing the display interval does not change the component's status indicators either.

You configure thresholds by navigating to **Status Summarizers**, right-clicking **Component Status Summarizer**, and clicking **Properties**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
    └ Site Settings
     └ Status Summarizers
```

You set the threshold period on the **General** tab and the threshold values on the **Threshold** tab. For more information, see the "Configuring the SMS Status System" section later in this chapter.

Note

The Component Status Summarizer reports a status message every time the status indicator for a component changes. You can configure **Status Filter Rules** to run a program to activate your alert system when this status message is processed. For more information, see the "When to Use Status Filter Rules" section later in this chapter.

The **Next started**, **Last started**, and **Last message** columns are times and dates. This information is displayed in the time zone for the site you are viewing. For example, if you are viewing Component Status for site CHI, the three columns show times and dates in Central Time.

Site System Status

The Site System Status summary is a high-level view of the health of all of the site systems at a given site. The details pane includes one row for each storage object. Table 14.4 describes the data reported for storage objects.

Table 14.4 Site System Status Details Pane Columns

Column name	Data type	Description
Status	State	Health of the storage object: OK, Warning, or Critical.
Site system	State	Name of the site system containing the storage object.
Role	State	SMS site system role performed by the site system: client access point, distribution point, SQL Server, software metering server, component server, or site server.
Storage object	State	Path to the storage object if the storage object is a directory that contains files or the name of the database or transaction log if the storage object is a database or transaction log.
Total	State	Maximum amount storage space, measured in bytes, that the storage object can contain.
Free	State	Amount of free storage space, measured in bytes, for the storage object.
% Free	State	Percentage of free storage space available on the storage object.
Down since	State	Displays the date and time when the storage object first became inaccessible. The storage object is considered down if the site server failed to connect to the storage object due to network or security problems.

To refresh this data, in the SMS Administrator console, right-click **Site System Status** and select **Refresh**.

You launch the Status Message Viewer by right-clicking a site system and selecting **Show Messages**. The messages you view are the ones reported by all of the components running on the site system you select. Be careful with this option. If there are many components running on the site system, there might be a large number of messages.

The status indicator in the **Status** column represents the health of each storage object. The status indicator is set based on whether or not free storage space on the storage object was below the Warning or Critical threshold the last time the Site System Status Summarizer polled the storage object. You view the thresholds set for a storage object by right-clicking a storage object in the details pane and clicking **Properties**. For example, if the Warning threshold is 1024 KB and the Critical threshold is 512 KB, the storage object's status indicator is set to Warning when the free storage space ranges from 513 to 1024 KB, and set to Critical when the free storage space is 512 KB or less.

You configure the thresholds by navigating to **Status Summarizers** in the SMS Administrator console, right-clicking **Site System Status Summarizer**, and clicking **Properties**.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
    └ Site Settings
     └ Status Summarizers
```

You set the threshold values on the **Threshold** tab. For more information, see the "Configuring the SMS Status System" section later in this chapter.

If the Site System Status Summarizer cannot connect to the storage object to determine the **Total** and **Free** storage space, a storage object's status indicator is set to Critical. Network or security problems might cause the Site System Status Summarizer to fail to connect. When the status indicator is set to Critical due to network connection problems, the **Down since** column contains the time and date the connection problems first occurred. This time and date appears in the time zone for the site you are viewing. For example, if you are viewing Site System Status for your NYC site, the column shows times and dates in Eastern Time.

Note

Every time the status indicator for a storage object changes, the Site System Status Summarizer reports a status message. You can configure the Status Filter Rules to run a program to e-mail you or activate your alert system when this status message is processed. For more information, see the "When to Use Status Filter Rules" section later in this chapter.

The per-site level of site status

The per-site level of site status contains a rolled-up status of the Component Status and Site System Status for a given site. The details pane includes one row for each category of site status: Component Status and Site System Status. Table 14.5 describes the data reported in the details pane.

Table 14.5 Per-site Level of Site Status Details Pane Columns

Column name	Data type	Description
Status	State	Overall health of the category of site status for the site: OK, Warning, or Critical.
Category	State	Category of site status: Component Status or Site System Status.
Summary date	State	Last date that the category of site status for the site was updated.

To refresh this data, right-click the site you are viewing in the console tree and click **Refresh**.

The status indicator for each of the categories is a roll-up of the status indicators in the status summaries for that category of site status. For example, the status indicator for Component Status is set to OK while the status indicators for all of the components at that site are set to OK. The status indicator for Component Status is set to Warning if one or more status indicators of the components are set to Warning, and if none of the status indicators for the components are set to Critical. The status indicator for Component Status is set to Critical if one or more status indicators for the components are set to Critical.

In the SMS Administrator console, the Component Status and Site System Status folders are overlaid with the status indicator for each category. Also in the console tree, the icon representing the site you are viewing is overlaid with the status indicator from whichever category is more critical. For example, if the status indicator for Component Status is set to Warning and the status indicator for Site System Status is set to Critical, the icon representing the site is set to Critical.

The summary date changes whenever any of the data changes for the category of site status. This happens, for example, whenever the Site System Status Summarizer runs a polling cycle to determine the total and free storage space on all the storage objects at a site. The date also changes whenever the Component Status Summarizer receives a status message from a component, and the count of Error, Warning, or Informational status messages is updated for that component. This time and date appears in the time zone for the site you are viewing. For example, if you are viewing the NYC site, the column shows dates and times in Eastern Time.

Note

The summary date is particularly important when you are monitoring a child site's status from the parent site. The summary date for a child site is always slightly old due to the delays in replicating the summary data up the site hierarchy. You can adjust the priority at which status summaries are replicated up the hierarchy. For more information, see the "Configuring the SMS Status System" section later in this chapter. If you find an extremely old summary date for a status summary of a child site, the parent site has not received a status summary from that site in a while, and there could be problems in the replication process.

The all-sites level of site status

The all-sites level of Site Status contains a rolled-up status of the Component Status and Site System Status of the current site and all of the sites below it in the hierarchy. The details pane includes one row per site. Table 14.6 describes the data reported in the details pane.

Table 14.6 All-sites Level of Site Status Details Pane Columns

Column name	Data type	Description
Status	State	Overall health of this site: OK, Warning, or Critical.
Site	State	Site code and site name for this site.

(continued)

Table 14.6 All-sites Level of Site Status Details Pane Columns *(continued)*

Column name	Data type	Description
Version	State	Version of SMS installed on this site, including the service pack, if one is installed.
State	State	State of this site.
Error	Count	Total number of Error status messages reported by all server components at this site during the display interval.
Warning	Count	Total number of Warning status messages reported by all server components at this site during the display interval.
Informational	Count	Total number of Informational status messages reported by all server components at this site during the display interval.
SMS DB % Free	State	Percentage of free storage space available for the site database.
Trans. Log % Free	State	Percentage of free storage space available for the site database's transaction log.

To refresh this data, right-click the site you are viewing in the console tree and click **Refresh**.

Note

The Site Hierarchy console tree represents the site as a hierarchy of console nodes, but the Site Status console tree item in the SMS Administrator console represents the site hierarchy as a list of console nodes. The Site Status console tree provides no information as to where a site is in the hierarchy. You should obtain that information from the Site Hierarchy console tree.

The status indicator for each of the sites is the rolled-up status indicator produced at the per-site level for each of the sites. In the console tree, the **Site Status** folder is overlaid with the status indicator from whichever site is the most critical. For example, the **Site Status** folder is overlaid with an OK status indicator as long as the status indicators are set to OK for all of the sites in the hierarchy. The **Site Status** folder is overlaid with a Warning status indicator if one or more status indicators for the sites are set to Warning and none of the status indicators for the sites are set to Critical. The **Site Status** folder is overlaid with a Critical status indicator if one or more status indicators for the sites are set to Critical.

You can determine the health of your site hierarchy at a glance by examining the status indicator applied to the **Site Status** folder in the SMS Administrator console. If the **Site Status** folder indicator is OK, all of the components and storage objects in your site hierarchy are OK. If the **Site Status** folder indicator is not OK, you can then browse through the console tree to determine which components and storage objects are not OK.

The **Errors**, **Warnings**, and **Info** columns are the sums produced from the Component Status summaries for all of the sites. You can view these counts for different display intervals. For more information, see the "Component Status" section earlier in this chapter.

Note

The display interval at the all-sites level of Site Status does not change when you change the display interval for the Component Status summary for a given site.

You launch the Status Message Viewer by right-clicking a site and clicking **Show Messages**. The messages you view are the ones reported during the display interval by all of the server components running at the site you selected. Be careful with this option. Depending on the display interval you select, there might be a large number of messages.

The **SMS DB %*Free and Trans. Log*% Free** columns are obtained from Site System Status for each site.

Package Status

Package Status provides a summary report of the health of packages and distribution points in your site. In many organizations, administrators distribute multiple packages concurrently to multiple destinations. Package Status allows you to monitor when packages arrive at distribution points. A package must arrive at a distribution point before a client can access, install, or run an advertisement. All dates in Package Status are displayed in the time zone of the computer running the Administrator console.

Package status provides three levels of status information details:

- Summary status for all packages in all sites
- Details for a specific package in all sites
- Details for a specific package in a single site

Summary Status for All Packages in All Sites

When you select Package Status, SMS displays a summary of the status for each package. This summary includes the status of all distribution points that the package has been assigned to in all sites. All values displayed include the current site and any child sites. Table 14.7 shows the information displayed in the details pane when you select **Package Status** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ System Status
   └ Package Status
```

Table 14.7 Package Status Details for All Sites

Column name	Description
Name	Shows the name you assigned the package when you created it in the SMS Administrator console.
Source Version	Identifies the current version of the package source files, as defined by the originating site. If you are viewing package status at a child site, the version displayed is the one currently in use at the child site. When you specify a new package source directory in the **Package Properties** dialog box or use the **Update Distribution Points** option from the **Package**, **Task** menu for a specific package, SMS increments the version number. This assists you in determining whether a distribution point has the latest version of the package source.
Version Date	Identifies the date and time when the version in the **Source Version** column was created.
Targeted	Displays the total number of distribution points, including child sites, that are specified to have a copy of the package. A distribution point remains targeted until it is specified for removal.
Installed	Shows the total number of distribution points to which the current source version of the package has been previously copied. A distribution point is considered installed until an update, refresh, or removal operation is specified.
Retrying	Displays the total number of distribution points for this package that have had at least one failure during an installation or removal operation, but that have not yet exceeded the number of retries allowed and are currently in an Installation Retrying or Removal Retrying state.
Failed	Displays the total number of distribution points for this package that have exceeded the number of retries allowed during an installation or removal operation, and that are currently in a Installation Failed or Retry Failed state.
Source Site	Shows the site where the package originated.
Size	Displays the size of the package source directory.
Compressed Size	Displays the size of the compressed version of the package source directory. The compressed size is not calculated unless creation of a compressed copy is specified in the **Package** properties or unless the package is targeted to a distribution point in a child site.
Package ID	Shows the internally assigned identifier for the package. The first three digits are the site code of the originating site.

Details for a Specific Package in All Sites

You can view summary information relative to the sites where each specific package has been sent. The summary that displays when you select ***<package name>*** includes information about the package with totals for all sites in your site hierarchy.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Package Status
  └ package name
```

Table 14.8 lists the information that SMS displays in the details pane when you select a specific package.

Table 14.8 Details for a Specific Package in All Sites

Column name	Description
Site	Displays the site code and site name for each site that has at least one distribution point specified to have a copy of the package.
Source Version	Identifies the version of the package source currently in use at individual SMS sites. When you specify a new package source directory in the **Package Properties** dialog box or use the **Update Distribution Points** option from the **Package**, **Task** menu for a specific package, SMS increments the version number. This assists you in determining whether a distribution point has the latest version of the package source.
Targeted	Shows the total number of distribution points in the sites that are targeted for this package.
Installed	Shows the total number of distribution points in the sites that have successfully installed the current source version of the package.
Retrying	Displays the total number of distribution points in the sites that have had at least one failure during an installation or removal operation, but that have not yet exceeded the number of retries allowed and are currently in an Installation Retrying or Removal Retrying state.
Failed	Displays the total number of distribution points in the sites that have exceeded the number of retries allowed during an installation or removal operation, and that are currently in an Installation Failed or Retry failed state.
Summary Date	Displays the most recent date and time when a change in package status for the sites was reported.

Details for a Specific Package in a Single Site

After you view package and distribution point information relative to all sites in your hierarchy, you might want to view the package status for a package in a particular site. You view package details for a single site by selecting ***<site code - site name>*** under a package in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
   └ Package Status
    └ package name
     └ site code - site name
```

Table 14.9 displays the information SMS displays in the details pane when you select ***<site code - site name>*** for a specific package.

Table 14.9 Details for a Specific Package in a Single Site

Column name	Description
Distribution Point	Displays the computer name for each distribution point in the selected site specified to have a copy of the package.
Source Version	Identifies the version of the package source currently installed on the distribution point. This column is blank before the package is installed on the distribution point.
State	Indicates which, if any, operation the Distribution Manager component in the selected site has pending or currently underway for the distribution point, or if the distribution point is in an idle and ready state. Possible distribution point states are: Installed. Installation pending. Installation retrying. Installation failed. Removal pending. Removal retrying. Removal failed.
Last Copied	Displays the date and time when the current package source version for the selected site was last successfully copied to the distribution point.
Type	Shows the site system of the distribution point.
Path	Displays the path that the Distribution Manager component selected to store this package on the distribution point. Note that the syntax used in the details pane is appropriate for the type of operating system in use.
Summary Date	Shows the most recent date and time that any site has indicated a change in status for the selected distribution point in the selected site.

If you launch the Status Message Viewer by right-clicking an item in any of the Package Status details panes and clicking **Show Messages**, the messages you view are the ones relevant to the package you selected.

Advertisement Status

Advertisement Status provides a summary report of the total number of clients that have received and run an advertisement. Table 14.10 shows the information SMS displays in the details pane when you select **Advertisement Status** in the SMS Administrator console.

Table 14.10 Advertisement Properties Summary

Column name	Description
Name	Displays the name the administrator gave the advertisement when it was created.
Failures	Displays the total number of users, client computers, or both in the site that experienced an error processing the advertisement or its associated package, or that attempted to run the advertised program but failed.
Programs Started	Displays the total number of users and/or client computers in the site that successfully ran the advertised program.
Program Errors	Displays the total number of users, client computers, or both that reported errors while running the advertised program. A program is considered in error when it produces either a non-zero exit code or an Install status MIF file with a **Failure** status attribute.
Program Success	Displays the total number of users, client computers, or both reporting that the advertisement ran successfully. A program is considered successful when it produces either an exit code of zero or an Install status MIF file with a **Success** status attribute.
Package	Displays the name of the package specified in the advertisement.
Program	Displays the name of the program specified in the advertisement.
Target Collection	Shows the target collection specified in the advertisement.
Available After	Shows the date, in local time, when the advertisement is made available to clients.
Expires After	Shows the date, in local time, when the advertisement expires and is no longer available to clients.
Advertisement ID	Displays the unique ID that SMS assigns to each package.

When you select a specific advertisement, SMS displays summary information that contains a per-site breakdown of the advertisement status as shown in Table 14.11.

Table 14.11 Advertisement Status Summary

Column name	Description
Site Type	Displays whether the site receiving the advertisement is a primary or a secondary site.
Site	Displays the site code and site name for the site receiving the advertisement.
Received	Displays the total number of users, client computers, or both in the site reporting successful receipt of the advertisement.
Failures	Displays the total number of users, client computers, or both in the site that experienced an error processing the advertisement or its associated package, or that attempted to run the advertised program but failed before the program being run.
Programs Started	Displays the total number of users and/or client computers in the site that successfully ran the advertised program.
Summary Date	Shows the most recent date and time that any site has indicated a change in status for the selected advertisement.
Program Errors	Displays the total number of users, client computers, or both that reported errors while running the advertised program. A program is considered in error when it produces either a non-zero exit code or an Install status MIF file with a **Failure** status attribute.
Program Success	Displays the total number of users, client computers, or both reporting that the advertisement ran successfully. A program is considered successful when it produces either an exit code of zero or an Install status MIF file with a **Success** status attribute.

If you launch the Status Message Viewer by right-clicking an item in any of the Advertisement Status details panes and clicking **Show Messages**, you view the status messages relevant to the particular advertisement.

Configuring the SMS Status System

You configure the SMS Status System from several locations within the SMS Administrator console. There are four important configuration areas:

- Status Component configuration
- Status filter rules
- Status summarizers
- Deleting status objects

The order in which you configure these items is not important.

Important

Do not attempt to configure the status system until you have used it for a period of time and you are confident that you fully understand the ramifications of changing the various default settings.

Status Reporting Configuration

You configure how components report status messages by navigating to **Component Configuration** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
    └ Site Settings
     └ Component Configuration
```

After you click **Component Configuration**, in the details pane, right-click **Status Reporting**, and then click **Properties**.

By using the **Status Reporting Properties** dialog box, you configure which status messages are reported to the Status Manager. You can also configure which status messages are logged to the Windows Event Log from this dialog box. For both reporting and logging, you can select the following:

- All milestones and all detail messages
- All milestones
- Error and warning milestone messages
- Error milestone messages
- Report detail messages on failure

Report detail messages on failure This feature is powerful. After a successful operation, components report milestone messages. Then, when a failure occurs, the component also reports the detail messages. Typically, you are interested in seeing detail messages only when you need to debug problems.

Status filter rules SMS status messages are continually generated by SMS components, so it is possible for your SMS hierarchy to become flooded with status messages. By creating and configuring status filter rules, you specify how status messages are processed at the site you are configuring. When the SMS status system processes a status message, there are five possible types of processing that might occur. Each type of processing results in the message either being discarded, or reported to one or more locations:

- SMS site database
- Windows Event Log

- Parent site
- Status summarizers
- Passed to an external user-specified program

Tuning Status Message Configuration with Status Filter Rules

You can tune and customize how SMS processes status messages by configuring filter rules. For example, you can configure the system to:

- Discard status messages from a component that is flooding the system with the same status message over and over again.
- Replicate error status messages to the parent site at High Replication priority and replicate other status messages at Low Replication priority.
- Not replicate status messages from the SMS Client to the parent site while other status messages are being replicated.
- Alert you via the **net send** command when a particular component reports an Error status message.
- Keep status messages associated with particular advertisement in the SMS site database for 30 days and keep other status messages for only seven days.

Note

Do not configure status filter rules until you are thoroughly familiar with the SMS status system.

Having proper filter rules can make a big difference in how your site performs. After you become familiar with the status system, you might want to tune the system to improve site performance. The following sections give explanations and examples that help you determine how to configure status filter rules for your site.

When to Use Status Filter Rules

While you observe the number and types of messages being reported through your site hierarchy, you have questions about how to write filter rules that give you the messages you need. Some questions you might ask are:

Which status messages do I never need to view? The Status Message Viewer displays the messages that are in the SMS site database. After you determine which messages you do not want to save and view, you can configure the status filter rules to not write these messages to the SMS site database and not replicate them to a parent site. This saves database space, processing cycles on the site server, and network bandwidth used in intersite replication.

Which status messages do I need to store in the SMS site database for a long time?
You might configure the status filter rules to keep certain messages for a long period of time, but to keep other status messages for a short period of time. For example, from **Advertisement Status**, you can determine which SMS clients received a particular advertisement by launching the Status Message Viewer to view the status messages generated when the clients receive the advertisement. If you want to keep a record of those clients for 60 days, you can configure the status filter rules to keep those messages in the SMS site database for 60 days.

Do I want to view status messages at a parent site that were generated at a child site?
If the answer is no, then consider configuring the status filter rules at the child sites to not replicate status messages to the parent site. This lowers the amount of processing done at both the child sites and the parent site and decreases the amount of network bandwidth used in intersite replication. You can prevent the status messages from being replicated and still allow the status summaries to be replicated. This provides you a summary view of information at the parent site. Then, when you detect a problem, you can connect to the child site to view the actual status messages.

Which status messages reported from secondary sites do I need to view? Because secondary sites do not have site databases or SMS Administrator consoles, you must configure status filter rules at secondary sites to replicate messages to the parent primary site if you want to view them. Although the Status Manager was designed to process several million status messages per day at a primary site that runs on a high-performance computer, a primary site that has many child sites might get overloaded by all the status messages coming from those child sites. This problem is especially important in child secondary sites because their messages can be viewed only at the parent primary site.

Note

Pay close attention to the status message load being sent from child sites. Status Manager contains performance counters to assist you. To view them, run the System Monitor on the site server and choose the **SMS Status Messages** object.

Configuring Status Filter Rules

You configure status filter rules on a per-site basis. To tune status message processing optimally for your site hierarchy, you might specify different filter rules for different sites. The Status Manager is the server component that processes each status message, one at a time, using the status filter rules. Each of these rules contains a list of criteria and a list of actions. When a status message matches the list of criteria, the Status Manager performs the actions specified by that rule. Status messages are processed against each rule, one rule at a time, in the order that the rules are displayed in the Status Filter Rules details pane in the SMS Administrator console.

The order of the status filter rules displayed in the SMS Administrator console is significant because the first rule always runs first. Then, the Status Manager proceeds down the list and processes each rule in the order it is displayed. If a status message matches a higher-priority rule, the Status Manager performs the actions specified, even if a lower-priority rule does not specify those actions. For example, if a matching higher-priority rule specifies that the status message should be written to the SMS site database, the Status Manager writes the message to the site database, regardless of what the lower-priority rules specify. If a matching lower-priority rule specifies an action that is not specified by a matching higher-priority rule, the Status Manager performs that action. In this way, higher-priority rules override lower-priority rules.

A possible status filter rule action is "Do not process lower-priority rules." If a status message matches a rule specifying that action, the lower priority rules are not processed for that status message. If a status message matches two rules that both specify an action, that action is only run once. For example, if Rule A and Rule B both specify the "Replicate to parent site" action, only one copy of status messages matching those rules is replicated.

You can change the priority of a rule by right-clicking the rule, pointing to **All Tasks** and then clicking **Increment Priority** or **Decrement Priority**. In addition to setting a priority, you can also enable or disable status filter rules by right-clicking the rule, pointing to **All Tasks** and then clicking **Enable** or **Disable**. To create and modify status filter rules, navigate to **Status Filter Rules** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
    └ Site Settings
     └ Status Filter Rules
```

Then, right-click **Status Filter Rules**, point to **New**, and click **Status Filter Rule**.Some choices on the **Status Filter Rule Properties** dialog box have drop-down boxes. When you click the drop-down box arrow, SMS queries the site database to determine the possible values. If you know exactly which value you want, you can type it into the box without clicking the arrow. For example, if you know that you want the rule to match status messages from the SMS Executive component, type SMS_EXECUTIVE into the **Component** box. Ensure that you type in the correct value. If you misspell a component name or other value, you are not prompted to correct it. Table 14.12 explains the various items that you can enter on the **General** tab to filter status messages.

Table 14.12 Status Filter Rules General Tab

Item	Explanation
Name	The name you assign the status filter rule. This name appears in the results pane. The name must be unique; two rules cannot have the same name.

(continued)

Table 14.12 Status Filter Rules General Tab *(continued)*

Item	Explanation
Source	The source that reports the status message. For example, SMS Server (server components), SMS Client (client components), or SMS Provider (SMS Administrator console and any other tools that exercise WBEM to change the SMS configuration).
Site Code	The site code of the site that reports the status messages.
System	The name of the computer that reports the status message. The system could be a site system or an SMS client computer.
Component	The name of the component that reports the status message. The component could be an SMS server component (service or thread), SMS client component, or a client of the SMS Provider such as the SMS Administrator console.
Message Type	The type of message (Milestone, Detail, or Audit).
Severity	The severity of the message (Informational, Warning, or Error).
Message ID	The numeric ID of the status message. You find the numeric ID of specific status messages through the Status Message Viewer or the **Event** column of the Windows Event Viewer.
Property Name	The optional property present in some status messages but not in others. For example, most status messages from the Distribution Manager component have a Package ID that identifies the package the message applies to. Alternatively, all Audit status messages have a **User Name** property that identifies the user who performed the action specified in the messages. To specify a property name, you must first specify a source.
Property Value	The value of the optional property specified by **Property Name**. For example, if you enter Package ID for **Property Name**, the **Property Value** might be NYC00001. If you select the box for **Property Name** but do not select the box for **Property Value**, then the rule matches all status messages that have the optional property associated with them, regardless of the value.

To complete a new status filter rule, you must set items on the **Actions** tab.

Table 14.13 Status Filter Rules Actions Tab

Item	Explanation
Write to the SMS site database	Specifies that status messages matching this rule should be written to the SMS site database. You must also specify how long the message should be kept in the database by adjusting the **Keep message for X days** value. The **Delete Status Messages** task under **Database Maintenance** in the SMS Administrator console uses the number of days listed in the status filter rules to determine which messages to delete and when.

(continued)

Table 14.13 Status Filter Rules Actions Tab *(continued)*

Item	Explanation
Report to the Windows Event Log	Specifies that the status messages matching this rule should be written to the Windows Event Log on the site server. See the "Using the SMS Status System with the Windows Event Log" section later in this chapter. When you select this option, SMS writes Milestone, Detail, and Audit messages to the Application Event Log.
Replicate to the parent site	Specifies that the status messages matching this rule should be replicated to the parent site, if there is a parent site. You should also specify the priority at which messages should be replicated with the **Replication priority** value.
Run a program	Specifies that each time a status message matches this rule, the Status Manager should run the program and command-line arguments specified in the **Program** edit box.
Do not forward to status summarizers.	Specifies that each time a status message matches this rule, the Status Manager should not forward a copy of this status message to any status summarizer components. You use this option to throw away messages that are flooding the system. For example, if a server component reports a particular Error status message every 10 seconds that is not relevant for your site, that error is not tallied by the Component Status Summarizer. A long series of these errors in a short time causes the Critical Threshold to be exceeded, and the component is flagged as Critical in the Component Status summary display. You can set up a filter rule to prevent a particular Error status message from being forwarded to the Component Status Summarizer, which prevents the threshold from being exceeded and the component being marked as Critical.
Do not process lower-priority status filter rules	Specifies that each time a status message matches this rule, the Status Manager should not process any lower-priority rules for that status message. This is a useful way to create rules to delete particular kinds of status messages. For example, if you are not interested in status messages from SMS_ INBOX_MANAGER, you could create a rule that matches all status messages from SMS_INBOX_ MANAGER that has no actions checked except **Do not forward to status summarizers** and **Do not process lower-priority status filter rules**. Place this rule at the top of the list. The **Do not process lower-priority status filter rules** action prevents the lower-priority rules from being processed.

Sample Status Filter Rules

Earlier in this chapter, you were introduced to five example status filter rules. This section provides instructions for creating each of these filter rules. Some of these examples might be directly applicable to a problem you face, and others might be less useful to you. By reading each of them, you gain a better understanding of how to use status filter rules for your organization.

If you decide to use some of these example rules at one of your sites, ensure that you examine the existing rules to see the status messages they match, the actions they perform, and their priority relative to your new rules. If your new rules are lower priority than existing ones, then the existing rules are processed before your new rules and might interfere with what you are trying to accomplish with the new rules. However, you can change the priority order.

To configure status filter rules, navigate to **Status Filter Rules** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
    └ Site Settings
     └ Status Filter Rules
```

Right-click **Status Filter Rules**, point to **New**, and click **Status Filter Rule**. Each of the examples below assumes that you know how to access the **Status Filter Rule Properties** dialog box.

► To discard status messages from a component that is flooding the system with the same status message multiple times

1. Create a new status filter rule.
2. On the **General** tab, in the **Name** box, enter **Throw away status message *X* from component *Y* on system <*computer name*>**, where *X* is the Message ID of the message you want to discard, and *Y* is the name of the component that is generating it.
3. Click **Site system**. Type the name of the computer the component is running on.
4. Click **Component** and enter the name of the component, *Y*. Click **Message ID** and enter in the Message ID of the status message, *X*.
5. Verify that no other items are selected. Click the **Actions** tab.
6. Click **Do not forward to status summarizers**. Click **Do not process lower-priority status filter rules**. Verify that no other items are selected, and then click **OK**.
7. In the Status Filter Rules results pane, right-click the new rule and select **Increment Priority** until the new rule is the first one in the list.

   This step ensures that Status Manager processes each status message against this filter rule first. Because the rule specifies **Do not process lower-priority status filter rules**, none of the other rules have the opportunity to write the message to the SMS site database, or replicate it to the parent site.

► To replicate error status messages to the parent site at high replication priority and replicate all other status messages at low replication priority

1. Create a new status filter rule. On the **General** tab, in the **Name** box, type **Replicate Error status messages at High priority**.
2. Click **Message Type** and select **Error**. Verify that no other items are selected.
3. Select the **Actions** tab. Click **Replicate to the parent site** and set the priority to **High**. Verify that no other items are selected.
4. Create a second new status filter rule. On the **General** tab, in the **Name** box, enter **Replicate all status messages at low priority**. Verify that no other items are selected.
5. Click the **Actions** tab. Click **Replicate to the parent site** and set the replication priority to **Low**. Verify that no other items are selected.
6. In the Status Filter Rules results pane, use the **Increment Priority** or **Decrement Priority** options until **Replicate Error status messages at High priority** is listed above **Replicate all status messages at Low priority**.

The first rule matches all Error status messages and causes Status Manager to replicate them at high priority. The second rule matches all status messages and causes the Status Manager to replicate the messages at low priority. Because the first rule is higher priority than the second, it overrides the second rule and causes Status Manager to replicate the status messages at high priority instead of low priority.

► To not replicate status messages from the SMS Client to the parent site, but replicate all other status messages

1. Create a new status filter rule.
2. On the **General** tab, in the **Name** box, type **Do not replicate SMS Client status messages**. Click **Source** and select **SMS Client**. Verify that no other items are selected.
3. Click the **Actions** tab. Click **Do not process lower-priority status filter rules**. Click any other options as appropriate, but do not select **Replicate to parent site**.
4. Create a second new status filter rule.
5. On the **General** tab, in the **Name** box, type **Replicate all other status messages**. Verify that no other items are selected.
6. Click the **Actions** tab. Click **Replicate to the parent site** and select a replication priority. Choose any other options as appropriate. Click **OK**.
7. In the Status Filter Rules results pane, use the **Increment Priority** or **Decrement Priority** options until the first new rule is above the second new rule.

The first rule matches all status messages from the SMS client and prevents Status Manager from replicating them. The second rule matches all status messages and causes Status Manager to replicate them at the priority you chose. Because the first rule is higher priority than the second, it overrides the second rule in the case of status messages from the SMS client computer and prevents Status Manager from replicating them.

If you had not specified **Do not process lower-priority status filter rules** in the first rule, the second rule would cause Status Manager to replicate SMS Client status messages. This nuance might not appeal to you, or it might complicate the other rules you want to define. An alternative way to express what this example seeks to accomplish is "Replicate status messages from the SMS server and SMS Provider, but do not replicate SMS client status messages."

► To replicate status messages from the SMS Server and SMS Provider, but not replicate SMS client status messages

1. Create a new status filter rule.
2. On the **General** tab, in the **Name** box type **Replicate SMS Server status messages**. Click **Source** and select SMS Server. Verify that no other items are selected.
3. Click the **Actions** tab. Click **Replicate to the parent site** and select an appropriate replication priority. Verify that no other items are selected.
4. Create a new status filter rule.
5. On the **General** tab, in the **Name** box, type **Replicate SMS Provider status messages**. Click Source and select SMS Provider. Verify that no other items are selected.
6. Click the **Actions** tab. Click **Replicate to the parent site** and select an appropriate replication priority. Verify that no other items are selected.

The first rule matches all status messages from the SMS Server and causes Status Manager to replicate them at the priority you chose. The second rule matches all status messages from the SMS Provider and causes Status Manager to replicate them at the priority you chose. It does not matter which rule is higher priority than the other, because they are mutually exclusive, as a status message cannot be from both the SMS Server and the SMS Provider. These two rules might appeal to you more than the first two example rules because these two do not involve the **Do not process lower-priority status filter rules** option.

► To alert you via the "net send" command when a particular component reports an Error status message

1. Create a new status filter rule.
2. On the **General** tab, in the **Name** box, enter **Alert me when the Inbox Manager component reports any Error status messages**.
3. Click **Component** and select **Inbox Manager** from the drop-down box. Click **Severity** and select **Error**. Verify that no other items are selected.
4. Select the **Actions** tab.
5. Click **Run a program**, and in the **Program** box, type the following:

   C:\Winnt\System32\Net.exe send %*sitesvr Component* %msgcomp running on computer %*msgsys reported the following error:* %msgdesc

   Replace *C:\winnt\system32* with the appropriate path to the system32 directory on your site server.

6. Verify that no other items are selected. Click **OK**.
7. In the Status Filter Rules results pane, select the new rule and right-click **Increment Priority** until the new rule is the first one in the list. This ensures that Status Manager processes this rule first, which prevents any of the existing rules from interfering with it. This rule causes any Error status messages processed by Status Manager that come from Inbox Manager to pop up as messages on the site server console.

The criteria that you specified on the **General** tab do not include a site code or a system name. You are alerted when Status Manager processes a status message from Inbox Manager running on any computer at any site, including child sites. To restrict the alerting to components from a specific computer or site, check the **System** or **Site Code** boxes on the **General** tab as appropriate.

► To keep status messages associated with a particular advertisement in the SMS site database for 30 days, but keep other status messages for only seven days

1. Create a new status filter rule.
2. On the **General** tab, in the **Name** box, enter **Keep status messages for Advertisement *X* in the database for 30 days**, where *X* is the Offer ID for the advertisement whose messages you would like stored for 30 days.

Note

Offer ID and Advertisement ID are the same.

3. Click **Source** and select **SMS Client**. Click Property name and select **Offer ID**. Click **Property value** and type in the Offer ID value. Verify that no other items are selected.
4. Click the **Actions** tab. Click **Write to the site database** and select **30** in the **Allow user to delete messages after: days** dialog box. Verify that no other items are selected.
5. Create a new status filter rule.
6. On the **General** tab, in the **Name** box, enter **Keep all other status messages in the database for 7 days.** Verify that no other items are selected.
7. Click the **Actions** tab. Click **Write to the site database and** select **7** in the **Allow user to delete messages after: days** dialog box.
8. Verify that no other items are selected.
9. In the Status Filter Rules details pane, use the **Increment Priority** or **Decrement Priority** options until the first new rule is above the second new rule.

   The first rule matches all status messages from the SMS client that are associated with the advertisement you specified and causes Status Manager to keep them in the site database for 30 days. The second rule matches all status messages and causes Status Manager to keep them in the database for only seven days. Because the first rule has higher priority than the second, it overrides the second rule in the case of status messages from the SMS client that are associated with your advertisement and cause Status Manager to keep them in the database for 30 days instead of seven days.

In the previous example, you had to specify a source to be able to specify **Property name** and **Property value**. This means that to keep messages from the SMS Server or SMS Provider that are associated with your advertisement for 30 days instead of seven, you must create an additional rule for the SMS Server and another for the SMS Provider. If you want to keep status messages from the SMS client that are associated with any advertisement for 30 days, you would leave **Property value** cleared.

Configuring Status Summarizers

Status Summarizer configuration enables you to determine the specific configuration of each status summarizer. You configure status summarizers by navigating to **Status Summarizer** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Site Hierarchy
   └ site code - site name
    └ Site Settings
     └ Status Summarizer
```

After you click **Status Summarizer**, you can select **Site Systems Status Summarizer**, **Component Status Summarizer**, or **Advertisement Status Summarizer** from the details pane.

You can disable any of the summarizers that might not be relevant at any particular time. For example, if you do not intend to send advertisements for an extended period of time, you might want to disable the Advertisement Status Summarizer. However, most of the time, you should leave your summarizers enabled because they provide valuable data.

Another feature includes both replicating a summary to the parent site and determining the replication priority. These are powerful features that help administrators at parent sites that monitor status at child sites. For component status and site system status, you can configure the thresholds SMS uses to determine when to report a warning or an error icon in the SMS Administrator console.

Deleting Status Messages

You enable status message deletion by navigating to **Tasks** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database (site code - site name)
  └ Sites
   └ site code - server name
    └ Site Settings
     └ Database Maintenance
      └ Tasks
```

Right-click **Delete Aged Status Messages**, and click **Properties**. The **Delete Aged Status Messages Properties** dialog box appears. You can both enable and schedule the deletion process. You must also configure the length of time that status messages remain in the SMS site database by using the **Actions** tab in **Status Filter Rules** feature. After you define the length of time that the status system should keep status messages, all messages older than the time you specified are deleted at the scheduled interval.

You can also delete status messages by using the Status Message Viewer or by creating a query to delete status messages.

Using the SMS Status System with the Windows Event Log

When SMS Status Manager reports a status message to the Windows Event Log, the message is translated into the columns available in the Windows Event Viewer as follows:

- **The Event** is always reported to the Application Log on the site server.
- **The Event <*Type*>** is the severity of the status message (Error, Warning, or Informational).
- **The Event <*Date*>** and **<*Time*>** are the local date and time the status message is reported as a Windows event on the site server, not the date and time the status message was reported originally. For a status message reported by a component at a child site, for example, the difference between the two dates can be substantial.
- **The Event <*Source*>** is the source of the status message (SMS Server, SMS client, or SMS Provider).
- **The Event <*Category*>** is the name of the component that reported the message, such as SMS_EXECUTIVE or the Available Programs Manager.
- **The Event <*ID*>** is the ID of the status message.
- **The Event <*User*>** is always "N/A."
- **The Event <*Computer*>** is the name of the site server, not the name of the computer that the component that reported the status message is running on.
- **The Event <*Description*>** is one of the following localized strings.

If the component is running on the site server at the current site: "On <*date, time, and time zone*>, component <*component name*> reported…"

If the component is running on a server at the current site other than the site server:
"On <*date, time, and time zone*>, component <*component name*> on computer <*computer name*> reported…"

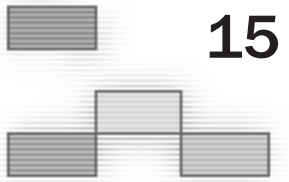
If the component does not belong to the current site: "On <*date, time, and time zone*>, component <*component name*> on computer <*computer name*> at site <*site code*> reported…"

- This date, time, and time zone specify the actual time the message was generated by the component.
- This is the date and time information that is most important to you in debugging your site.
- The **Event Data** is always grayed out.

To set up alerts similar to the ones used in SMS 2.0, you can configure SMS to send status messages to the Windows Event Log. Then, set an alert based on the criteria you specify.

CHAPTER 15

Backup and Recovery



To avoid the loss of critical data, you must plan and prepare for both backup and recovery operations. It is important that you are able to quickly recover Microsoft® Systems Management Server (SMS) 2003 sites and hierarchies with as little data loss as possible.

A total backup and recovery cycle consists of four major administrative tasks:

1. Planning for backup and recovery — done during the SMS deployment planning phase.
2. Preparing for recovery — done during the SMS deployment phase.
3. Backing up a site — done on a regular basis after deploying SMS.
4. Recovering a site — done as needed.

This chapter assumes that you have read and are familiar with Chapter 13, "Planning for Backup and Recovery," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide.*

This chapter does not describe backup and recovery issues related to upgrading to SMS 2003. For information about backup and recovery upgrade issues, see Chapter 14, "Upgrading to SMS 2003," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide.* For information about backup and recovery issues in a mixed-version hierarchy, see Chapter 6, "Understanding Interoperability with SMS 2.0," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

In This Chapter

- Planning for Backup and Recovery
- Preparing for Recovery
- Backing Up a Site
- Recovering a Site

Planning for Backup and Recovery

It is essential that you plan for backup and recovery as early as possible. The best time to plan for backup and recovery is during the SMS deployment planning phase.

If your site fails and you do not have a backup and recovery plan, you increase the time that the site server is not functioning, reducing the level and quality of services to clients.

If you do not have a backup and recovery plan, then:

- The risk of site failure is higher.
- The impact of failure on your site is greater.
- The recovery process is more complex, takes longer, and more data is lost during the process.

For information about backup and recovery concepts and backup and recovery planning, see Chapter 13, "Planning for Backup and Recovery," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide.*

Preparing for Recovery

You should prepare for recovery while your site is healthy, preferably as soon as your hierarchy is set up and configured. First, you plan your site hierarchy, incorporating recovery-related deployment requirements. Then, you install and configure your site. As data starts to flow in the site and between sites, you need to prepare each site in your hierarchy for recovery.

The main recovery preparation step consists of backing up your site, and then archiving the backup snapshot. This is described later in this chapter.

Designating Reference Sites

The SMS Site Repair Wizard uses reference sites to reclaim lost objects and determine appropriate serial numbers during the repair phase of a recovery operation. Any primary site that is lower in the hierarchy than a failed site can be a reference site. A reference site should exist at each tier of the hierarchy.

You can designate sites at the lowest tier of the hierarchy as reference sites. Do not use these sites for actual management. Use them only as a repository for replicated data from higher level sites.

For more information about designating reference sites, and about the SMS Site Repair Wizard, see the "SMS Site Repair Wizard" section later in this chapter.

Documenting Design Information, Configuration Data and Accounts Passwords

To correctly recover a site system, you must have all server configuration data available. You must configure the site system server exactly the way it was configured when it failed. If you do not, site recovery can fail. In this situation, it might be difficult for you to know why the recovery did not work.

To successfully recover a site, account passwords are required. For security reasons, in SMS 2003, accounts such as Client Push Installation Account, Site Address, site system connection accounts, and network access account are encrypted. As a result, when a site fails, these accounts are lost and cannot be retrieved during a recovery. During a site recovery, you will be prompted to re-enter these account passwords.

Because it might be impossible to obtain data from a failed system, it is important that you gather and document the necessary data while the system is healthy.

Document and store:

- The hierarchy design. Include parent-child relationships, primary and secondary site locations, and information about the SMS site database server.
- Site configuration data from each SMS site server in your hierarchy. Include information such as feature settings, site systems, and component settings.
- List of designated reference sites.
- Passwords for the Client Push Installation Account, Site Address Accounts, Package Access Accounts, Site System Connection Account, and Advanced Client Network Access Account.

Ensuring that this data is always available and current helps you in case a backup snapshot is not available, or in the event that there is no staff familiar with the hierarchy deployment.

Setting Up a Recovery Expert Web Site and Running the Recovery Expert

To run the Recovery Expert tool, you must first set up a Recovery Expert Web site on a computer running Internet Information Services (IIS) version 5.0 or later. If you allocate a server running an operating system in the Microsoft Windows Server™ 2003 family, then you must set the Active Server Pages Web Service Extension to **Allow**.

► To change the status of Active Server Pages to Allow

1. From the **Start** menu, point to **Programs**, point to **Administrative Tools**, and then click **Internet Information Services**.
2. In the left pane of the **Internet Information Services** console, click **Web Service Extensions**.
3. In the right pane, click **Web Service Extension – Active Server Pages**, and then click **Allow**. This changes the status of Active Server Pages to **Allow**.

► To set up the Recovery Expert Web site from the SMS CD

1. Insert the SMS 2003 product CD into the designated IIS server that will host the Recovery Expert Web site tool.
2. From the CD, run Autorun.exe.

3. In the **Systems Management Server 2003 Setup** dialog box, select **Recovery Expert**.
4. Finish the Microsoft SMS Recovery Expert Web Site Installation Wizard.
5. Note the URL displayed on the last page of the wizard so that you can refer to it later. Inform other SMS administrators about this URL so they can use it to access the Recovery Expert Web site, and to run the Recovery Expert tool.

Important

If you use the Microsoft IIS Lockdown tool (lislockd.exe) to increase security protection on a computer running IIS, apply it to the computer (using the SMS 2003-specific template) before setting up a Recovery Expert Web site on that computer.

For information about the role of the Recovery Expert in a site recovery operation, see the "Recovering a Site" section later in this chapter.

Security settings

The Recovery Expert requires that Internet Explorer be configured with medium security. In the **Internet Options** dialog box, on the **Security** tab, set security in either of the following methods:

- Set **Local intranet** security to medium.
- Set **Local intranet** security to high, add the Recovery Expert Web Site to the **Trusted sites** zone, and set the security of **Trusted sites** zone to medium.

When upgrading a server from Microsoft Windows® 2000 Server to a server in the Windows Server 2003 family, the upgraded server's default security permissions are more restrictive. These security settings will prevent the Recovery Expert from running on that server. After the upgrade, you must manually reconfigure the permissions. This applies whether the Recovery Expert was installed before or after the upgrade.

► To reconfigure security settings on a server upgraded to a server in the Windows Server 2003 family:

1. In Windows Explorer, select the following file: C:\Inetpub\wwwroot\SMSComponent\FormatMessageCtl.dll.
2. Right-click the file and select **Properties**.
3. In the **<file> Properties** dialog box, click the **Security** tab.
4. In the **Group or user names** list, select **Internet Guest Account**.
5. In the **Permissions for** list, ensure that **Allow** is selected for the **Read & Execute** permission.

► To run the Recovery Expert

1. In Internet Explorer version 5.5 or later, use the Recovery Expert Web site URL to access the Recovery Expert Entry Page.
2. Read the introductory content.
3. Select **Use The Recovery Expert** to start the Recovery Expert.

Preparing Recovery Steps

It is recommended that you generate the Recovery Expert recovery tasks list for the site ahead of time. If you prepare those recovery tasks while the site is healthy, there is no need to run the Recovery Expert later if the site fails and a recovery operation is required. You can generate the site's recovery tasks list in advance by running the Recovery Expert while the site is healthy. Print the site's recovery tasks list and store it in an accessible location.

Although it is always recommended that you generate the recovery tasks list in advance, it is not always effective. It is recommended that you run the Recovery Expert and generate the site's recovery tasks list in advance in the following circumstances:

- When a site is configured according to generic organization standards, and these configurations are not likely to change.
- When the current administrator of the site is not the administrator that designed the site. It makes sense for the designing administrator to prepare recovery answers in advance.
- When the administrator that administers a recovery process is likely to be an unskilled SMS administrator. In this case, the prepared recovery steps must be continually updated with any changes to the site.

Backing Up the Central Site's Control File

There can be many configuration changes on the central site, which will be hard or impossible to repeat if there is a need to recover the central site. Even if you are backing up the central site, it is important to frequently back up the central site's control file in between the regular site backups.

For more information about backing up the central site's control file, see the "Backing Up the Central Site" section later in this chapter.

Preparing a Test Lab for Recovery Tests

The best way to be fully prepared for a site recovery operation is to ensure that the site's recovery plan is adequate, and that administrators are familiar with the recovery process. After you develop a recovery plan for your site, it is recommended that you perform periodic recovery tests.

You can perform recovery tests in the test lab that was used for the hierarchy initial deployment tests. A test lab is ideal for recovery tests because it represents the production environment, and it contains collections, packages, and advertisements similar to the ones at the production environment.

By performing recovery tests at the test lab, you can identify problems with your recovery plan and refine it to ensure that it is possible to successfully recover as many objects as possible in case a site fails. However, if your organization does not maintain a test lab, or if you cannot access a remote test lab, you might consider setting up a local test lab for recovery tests.

Prepare the test lab for recovery tests as follows:

- In the test lab, use backup and archive procedures that are similar to the procedures used in the production environment. Ensure that the SMSbkup.ctl files at the test lab and in the production environment are similar. Ensure that if the AfterBackup.bat file is used in the production environment, then a similar file is used in the test lab.
- Designate reference sites in the test lab so they are similar to the designated reference sites in the production environment.
- Set up a Recovery Expert Web site to be used for the recovery tests, unless it is possible to use the Recovery Expert Web site that is set up in the production environment.
- Configure site security in the test lab so it is similar to the security configuration in the production environment

For more information about setting up a test lab, see Chapter 7, "The Pre-Planning Phase," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Backing Up a Site

It is critical to back up your site. Backing up sites in your hierarchy is the most important recovery preparation step to ensuring a successful recovery in case of a site failure. Although it is possible to recover sites without a backup snapshot, recovering a site with a backup snapshot ensures the least data loss and a less complex recovery process.

To ensure that backing up a site is as easy as possible, SMS provides the Backup SMS Site Server task, which is sometimes referred to as the SMS backup task. The SMS backup task creates a backup snapshot, which is a copy of the site data. The site's backup snapshot is stored at a location that you specify, referred to as the backup snapshot destination. Backing up your site on a regular basis is the main step when preparing for recovery.

Start backing up your site when you have finished configuring the site and that site is functioning properly. Schedule backup cycles to back up your site on a regular basis to ensure having a recent backup snapshot of the site.

Overview of Site Backup

An SMS site contains a large amount of data, which is mostly stored in the registry, in system files, and in Microsoft SQL Server™ databases. SMS operates properly only when the data in all these data stores is synchronized with each other. If you recover a site and you restore data from one data store but do not restore data from the other data stores, then the data will be out of synchronization, and you might corrupt the site's data.

Backing up only one data store, such as the SMS site database, is not sufficient as a backup strategy. You will not be able to recover a site using a partial site backup because the site's data will be out of synchronization. You must back up all of the site's data as a snapshot. Even if only one data store is corrupted, the entire backup snapshot must be restored as part of a site recovery operation.

There are several SMS systems that contain site data, but it is not necessary to back up all those systems. You can recover a site successfully if you back up a snapshot of the data from the following site systems in your site:

- The site server
- The SMS site database server
- The site provider server

There is no need to back up data from site systems such as distribution points, management points, reporting points, and server locator points. This is because if such site systems fail, then the site server can easily recreate them.

Ideally, every SMS site has more than one computer performing similar site system roles. If a client's default site system is unavailable, the client can reach another site system with a similar site system role and use it until you restore the client's default site system. From the client's perspective, regular operations are not interrupted.

It is not recommended that you back up the site's clients for following reasons:

- To properly back up an SMS client, the client services must be stopped. However, there is no reliable way to stop and start the client services. Stopping and starting the client services can potentially corrupt the data on the client's disk or in the backup snapshot. Backing up clients risks data integrity.
- Clients are too numerous. It is neither practical nor beneficial to back up and restore thousands of clients.
- The effect of losing client data is relatively small.

Backing up a site is automated in SMS 2003 by the integrated Backup SMS Site Server task. Use the SMS backup task to regularly back up data such as SMS files, registry keys, and configuration information from your site server, and from your SMS site database server or provider server, if necessary.

You can use the Backup SMS Site Server task to back up any site in your hierarchy. However, there are some unique issues associated with backing up the central site and backing up secondary sites. These issues are addressed later in this chapter.

The Backup SMS Site Server Task

The Backup SMS Site Server Task backs up all the site's data that is required for a site recovery, as a snapshot. The Backup SMS Site Server Task is one of the predefined maintenance tasks. The task is disabled by default. To use the task, you must configure it in the SMS Administrator console and perform some other steps.

When the backup task backs up your site, it relies on input. It reads the configuration settings that you specified in the backup task properties dialog box. These values determine when and how often the task runs. It also reads input from the SMSbkup.ctl control file. Another file that can participate in the backup cycle is the AfterBackup.bat batch file. You can edit these two files to customize the backup operation to fit your site backup needs.

SMSbkup.ctl Contains site-specific information that the backup task requires. This file contains the names of the files, registry keys, and databases that need to be backed up. It also contains commands that run during the backup operation to gather configuration information. The SMSbkup.ctl file contains tokens that it uses during run time, such as the SITE_BACKUP_DESTINATION token. When the backup task runs and uses the default SMSbkup.ctl file, it backs up all data necessary for recovery. You can customize this file to address specific backup needs of your site. SMSbkup.ctl is referred to as the backup control file.

AfterBackup.bat Allows you to archive the backup snapshot at the end of every backup operation. By default, the AfterBackup.bat file does not exist and therefore has no effect on the backup operation. You can create this file and add commands that run after the SMS backup task has finished.

By default, the Backup SMS Site Server task does not back up the following:

- Remote Tools, Network Monitor, Package Automation Scripts, or WBEM. These components are not backed up because they are easy to re-install:

Note

In SMS 2003, software metering data is integrated with the rest of the site's data. There is no need to separately back up software metering data.

- Custom SMS files, such as custom SMS Administrator console files or custom MOF files, unless they are stored in directories that are being backed up.
- Any files related to site system roles that are set up on the site server. For example, the files related to a distribution point, or to a client access point (CAP).

The Effect of the Backup Task

When the Backup SMS Site Server task runs, it interferes with regular site server activity. To properly back up a site, the Backup SMS Site Server task stops the following basic site services:

- SMS_SITE_COMPONENT_MANAGER
- SMS_EXECUTIVE
- SMS_SQL_MONITOR

Without these services running on the site server, data arriving from clients is not processed. Also, you cannot perform some regular site operations. For example, you cannot:

- Troubleshoot client computers.
- Advertise new programs.
- Distribute new packages.
- Create new software licenses.

However, you can view reports.

In general, it is recommended that you do not make any site configuration changes or initiate any site activity while the SMS backup task runs. This is because changes are processed only after the backup operation is completed. Any site activity that you initiate is likely to slow down the backup operation.

When the backup operation is completed, the backup task restarts the basic site services that were stopped, and the site server returns to the state that it was in before the task started.

The backup task running on the site server has very little effect on clients. Clients do not interact directly with the site server for most of their activities. They continue to interact with site systems such as client access points and distribution points, which are not affected by the backup operation. However, the site server does not process status messages and inventory data from clients until after the backup operation is completed.

How the Backup SMS Site Server Task Works

The SMS_SITE_BACKUP service is the service that runs on the site server to accomplish the backup task operation. The SMS backup task runs under the SMS Service Account or the site computer account. Those accounts must have **Read** and **Write** permissions to the folder that you plan to back up the site to.

At its scheduled time, the SMS_SITE_BACKUP service starts a backup cycle. During the backup cycle, the service performs some initial steps, and then backs up data from the site server. It then backs up data from the SMS site database server and from the provider server, if either is set up on a computer other than the site server.

Using the default SMSbkup.ctl file, the backup service performs the following steps during a backup cycle:

1. The backup service sets the value of the SITE_BACKUP_DESTINATION token, in which it will store the backup snapshot, as follows:
   - It reads the value of backup destination specified in the **Backup SMS Site Server Properties** dialog box in the SMS Administrator console.
   - It uses the value of backup destination as follows:
     SITE_BACKUP_DESTINATION=*backup destination*\<*site code*>Backup
     (this directory structure allows multiple sites to share the same backup destination)

Caution

You must ensure that there is sufficient disk space to store a backup snapshot of the site at the backup destination.

2. The backup service verifies that the following initial requirements are met:
   - That the SMS_SITE_BACKUP service has full control over SITE_BACKUP_DESTINATION, so it can recreate the backup destination folder and copy files to that folder.
   - That the SMS backup control file, SMSbkup.ctl, is valid and has no syntax errors.

If any of the above requirements is not met, the backup service logs an error message to the backup log file and stops the backup operation.

3. The backup service creates the SITE_BACKUP_DESTINATION folder.
   If the SITE_BACKUP_DESTINATION folder exists (it typically contains the backup snapshot from the previous site backup), then the backup service does the following:
   - Removes the SITE_BACKUP_DESTINATION folder with its entire content.
   - Recreates the SITE_BACKUP_DESTINATION folder. If the backup destination is on the local server, then SMS configures the SITE_BACKUP_DESTINATION folder with access to the Administrators group. If the backup destination is on a remote system, then the SITE_BACKUP_DESTINATION folder inherits its access rights from the backup destination folder.
4. The backup service defines all tokens listed in the [Tokens] section in the SMSbkup.ctl file.
5. The backup service performs all commands that are listed in the [Stop] section in the SMSbkup.ctl file. By default, the backup service stops the following services:
   - SMS_SITE_COMPONENT_MANAGER
   - SMS_EXECUTIVE
   - SMS_SQL_MONITOR
6. The backup service performs all commands that are listed in the [Tasks] section in the SMSbkup.ctl file. By default, this includes the following:
   - Backing up SMS files.
   - Backing up registry keys.
   - Backing up the SMS site database.
   - Running tools to collect site configuration data, and then backing up that data.
7. The backup service performs all commands that are listed in the [Start] section in the SMSbkup.ctl file. By default, the backup service restarts the services that were stopped in step 5.
8. The backup service runs the AfterBackup.bat batch file if both of the following are true:
   - The file exists in the SMS\inboxes\smsbkup.box folder.
   - The backup task was successful.
9. The backup service attempts to save the backup log file, SMSbkup.log, at the SITE_BACKUP_DESTINATION. It saves the log file if both of the following are true:
   - A log file was created.
   - The SITE_BACKUP_DESTINATION folder was successfully created (even if the task failed at a later point). If this condition is not met, then the backup task preserves the previous backup log file.

Content of the Backup Snapshot

The SMS backup service stores the site data in files and creates the backup snapshot according to what you specified in the backup control file. The backup snapshot created with the default control file includes SMS files, registry keys, databases, and configuration information.

If the files in the backup snapshot follow the naming convention described in the "Using SMSbkup.ctl to Control the Backup SMS Site Server Task" section later in this chapter, then you can identify the content of each file in the backup snapshot. This is useful if you need to locate a specific file in the backup snapshot.

Backing Up a Site Using the Backup SMS Site Server Task

After a site is configured and running smoothly, start performing regular site backups. To successfully back up your site using the SMS backup task, you need to perform some manual tasks. Some tasks you perform before running the backup task for the first time, other tasks you perform each time the backup task runs, and a few tasks you perform on an infrequent basis.

To Prepare for the Backup Task

By default, the SMS backup task is disabled. To start backing up your site on a regular basis, you need to enable the task and perform the following additional steps:

1. Customize the backup control file, SMSbkup.ctl, if needed.
2. Create and customize the backup batch file, SMS\inboxes\smsbkup.box\AfterBackup.bat, to archive the backup snapshot, or to perform any other post-backup operations.
3. Enable, configure, and schedule the SMS Backup Site Server task.
4. Enable logging for the SMS Backup Site Server task.

Tasks to Perform Each Time the Backup Task Runs

When the backup task is configured to run on a regular basis, there are manual steps that you need to perform every time the task runs:

1. To reduce the risk of corrupting the backup snapshot, before backup starts, verify that the site is healthy.

   Review status messages, and log files if necessary to ensure that the site is properly functioning. To check the status of the SMS site database, you can schedule the following Database Consistency Check (DBCC) tests in SMS, or in SQL Server. On large, busy sites, the database tests might take a significant amount of time to be completed, so it is recommended that you schedule them to run on an opposite cycle from backup:

   - **DBCC** checkdb
   - **DBCC** checkcatalog

   To find out if the **DBCC** tests encountered any problems, you must read the **DBCC** output file.

2. After backup has finished, verify that the backup operation was successful.
3. Manually back up any custom SMS-related files that are not backed up automatically by the backup task. For example, custom SMS Administrator console files (.msc files,) custom MOF files (such as SMS_def.mof), and any Supplemental Reports that are stored at reporting points under \Reporting Point\Supplemental.

Important

If the site server is configured with the Advanced Security mode and SQL Server is running on an operating system in the Windows Server 2003 family, then you need to manually gather configuration information from SQL Server. For more information about this issue, see article number 316360 in the Microsoft Knowledge Base at http://support.microsoft.com.

If the software update management feature is used, it is very important to back up the Definitive Software Library. The Definitive Software Library is valuable because it contains package source folders of software updates that have already been downloaded, tested and authorized for distribution. In addition, it contains the list of authorized updates for each software update package (in the Patchauthorize.xml file). If the package source folders for software updates are not backed up, you will need to use the Distribute Software Updates Wizard to rebuild those folders after recovering a site. For more information about recovering software update packages, see the "Recovering a Site" section later in this chapter.

To automate the backup of any files that are not backed up by default, you can do either of the following:

- Store custom files under directories that are automatically backed up, for example, the SMS\Bin or the SMS\Inboxes directories.
  -Or-
- Add **file** commands where <*source*> is the path to the custom files to the SMSbkup.ctl file.

It is not necessary to manually back up data from site systems such as distribution points, management points, reporting points, and server locator points that are set up on the site server. This is because the site server can easily recreate them.

4. Document account passwords. For security reasons, SMS 2003 encrypts accounts such as the Client Push Installation Account, Site Address account, Package Access Accounts, site system connection accounts, and network access accounts. You need to document these account passwords so you can re-enter them during a site recovery operation.
5. Archive the site's new backup snapshot (if you have not used AfterBackup.bat to do so automatically) to a removable media.
6. Store the removable media in a secure location.

Tasks to Perform on an Infrequent Basis

Although the backup task runs on a regular basis, there are a few manual steps that you need to perform on an infrequent basis:

1. Perform a live recovery test, using the site's backup snapshot.
2. Manually back up site data that the backup task does not back up. The frequency of such backups depends on the frequency of updates to that data.
   - Whenever there is a change to the password of accounts such as the Client Push Installation Account, Site Address account, site system connection accounts, or network access account, you should note that change. This is important so you can re-enter these passwords during a site recovery operation.
   - If you use system tools to customize SMS site security, then you need to back up the security data of the site.
   - If you store prepared recovery tasks from the Recovery Expert, then whenever there are changes to the site configuration, you should rerun the Recovery Expert with updated answers. The resulting recovery tasks need to be backed up.

Using SMSbkup.ctl to Control the Backup SMS Site Server Task

The backup control file, SMSbkup.ctl, is an ASCII text file located on the site server at SMS\Inboxes\Smsbkup.box\.

When the backup task runs, it reads this file and performs the commands in the order that they appear in the file. The SMSbkup.ctl file drives the Backup SMS Site Server task and determines what data is backed up from the site and where it is stored at the backup snapshot destination.

The default SMSbkup.ctl file works well for common backup scenarios, but if your site has specific backup requirements, you can customize the file. To ensure a successful recovery, you must at least back up all the data that SMSbkup.ctl backs up by default. You can add custom backup tasks specific to your site, but do not discard any backup tasks included in the default SMSbkup.ctl. The only exception to this is if, for performance reasons, you choose to prevent the backup of one or more inboxes.

Not backing up specific inboxes

Using the default backup file, SMS backs up all subfolders under the \SMS\inboxes folder, some of which can contain a very large number of files. To increase the performance of the backup task, you might prefer not to back up a few specific inboxes, despite the expected data loss if the site fails. When you reduce the number of inboxes that are backed up, the backup task runs faster, reducing the site server's downtime. Also, the resulting backup snapshot is smaller, saving disk space.

Most inboxes must be backed up. Carefully evaluate the importance of data in each inbox, because data for some inboxes cannot be recreated on the recovered site, or recreating that data might be difficult or time consuming.

There are only a few Inboxes that the site can be successfully recovered without. You can choose to prevent backing up the following inboxes:

- Colfile.box — Hardware inventory data files waiting to be processed by the Inventory Data Loader. This data is regenerated at the first hardware inventory resynchronization cycle after recovering the site.
- Dataldr.box — Inventory MIF files waiting to be loaded into the site database by the Inventory Data Loader. This data is regenerated at the first hardware inventory resynchronization cycle after recovering the site.
- Ddm.box — Data discovery records waiting to be loaded into the site database by the Discovery Data Manager. No need to save it; Site System DDR would be regenerated seven days after recovering the site.
- Inventry.box — Hardware inventory data files received from the CAP, and from lower level sites. Those files are waiting to be processed into MIF files by the Inventory Processor. This data is regenerated at the first hardware inventory resynchronization cycle after recovering the site.
- Invproc.box — Hardware inventory data files that require history to be processed by Inventory Processor. This data is regenerated at the first hardware inventory resynchronization cycle after recovering the site.
- OfferSum.Box — Replication files from child site, and status message files waiting to be processed. On secondary sites, this folder also contains advertisement summarization files. After site recovery, most of the status messages are lost. The replication files from the child site will be sent again and should be recovered.
- Sinv.box — Software inventory data files received from lower level sites, waiting to be processed into the site database by the Software Inventory Processor. This data is regenerated during an inventory resynchronization cycle after recovering the site.
- Statmgr.box — Status message files received from lower level sites and waiting to be processed into the system's event viewer by Status Manager. Some of those status messages are regenerated by the appropriate components, but most are not.

You can prevent SMS from backing up inboxes by modifying the backup control file. Replace the **file** command that backs up the inbox folder with individual **file** commands for each inbox subfolder that you want to back up. Leave out the inbox subfolders that should not be backed up.

Caution

The SMS backup task is designed to back up specific data in an SMS site. Although you can use this backup task to back up other data, such backups can cause complications that are not associated with backing up the intended data from an SMS site.

Guidelines for Customizing SMSbkup.ctl

Before making any changes to the original SMSbkup.ctl file, save a copy of the file in case you need it later.

The information in the backup control file is organized into four sections:

1. [Tokens] — A list of tokens and their values that the backup task uses as variables when it runs.
2. [Stop] — A list of processes that the backup task stops before it starts to back up data.
3. [Tasks] — A list of backup commands targeted at different types of data at the site.
4. [Start] — A list of processes that the backup task starts after the backup task has been completed.

To ensure that the backup task produces a valid backup snapshot, carefully follow the modifications guidelines for each section, do not modify the order of the sections, and edit only the designated areas in the file.

[Tokens]

Several required tokens are predefined, and you cannot modify them. The default backup control file contains a list of all predefined tokens. The default structure of the backup snapshot destination folder allows multiple sites to share the same backup location.

You can customize this section by appending new tokens to it and assigning values to them as follows:

<MyToken>=<*token value*>

To reference a token, enclose it with the % character, as in ***%MyToken%***. You can use environment variables as tokens.

[Stop]

By default, the following basic services are stopped:

- %*SITE_SERVER*%\SMS_SITE_COMPONENT_MANAGER
- %*SITE_SERVER*%\SMS_EXECUTIVE
- %*SITE_DB_SERVER*%\SMS_SQL_MONITOR

You cannot modify this default behavior.

You can customize this section by appending the following commands:

- To stop a Windows service that is installed on the backed up site server, use

  service <*service name*>

  Where <*service name*> is a full service name without the path to the executable file. All running instances of <*service name*> are stopped.

  To stop a service on a remote computer, type the service name as follows:

  \\<*machine name*>\<*service name*>

  This command stops all instances of the <*service name*>. When you add a service to this section, consider the effect of this service being stopped during the backup operation.

The backup task attempts to stop a service only if that service is installed and it is running on the backed-up site server at the time that the backup task starts. The backup task restarts these services after the backup operation has been completed.

- To stop a Windows executable file that is running on the backed up site server, use

  exec <*executable name*>

  Where <*executable name*> is the file name of the executable file that you want stopped.

  You cannot stop an executable file that is running on a remote computer.

  You might not be able to stop an executable file properly. In some cases, when you stop an executable file, the operating system resources used by that executable file are not released. Therefore, use the **exec** command to stop an executable file only if it absolutely cannot be running during the backup operation and only if you cannot stop it using the **service** command.

- To pause the backup operation for a specified number of seconds, use

  sleep <*seconds*>

  Where <*seconds*> is a numeric value that specifies the number of seconds that the backup operation pauses. The maximum value for <*seconds*> is 900 (15 minutes).

[Tasks]

The default control file contains commands that back up all of the site data required for recovery. This includes SMS files, registry keys, configuration data, and the SMS site database.

All backup tasks in this section implicitly create the specified destination folder, if it does not already exist.

You can customize this section by appending the following commands.

- To back up a file object, use:

  file <*source*> <*destination*>

  Where <*source*> is a file or a folder to be backed up, and <*destination*> is the destination folder that the backup task copies the <*source*> object to. (If <*source*>is a folder, that folder is recursively backed up.)

  You can use the * and the ? wildcards when you specify <*source*>, as follows:

  - * replaces a variable length string.
  - ? replaces a one character string.

  Add **file** commands to back up custom SMS-related files, such as MOF files and custom SMS Administrator console files, that are not stored in one of the directories that the backup task automatically backs up.

  Any of the following conditions cause this command to fail:

  - The <*source*> folder does not exist.
  - The backup task cannot read the <*source*> folder.
  - The backup task cannot write to the <*destination*> folder.

- The <*source*> and <*destination*> directories are identical.
- The <*destination*> folder is a child folder of the <s*ource*> folder.

- To back up a registry object, use:

  reg <*source*> <*destination*>

  Where <*source*> is a registry key string, and <*destination*> is the destination file that the backup task copies the registry key to. To back up a registry key on a remote computer, prefix the registry key with a computer name in a Universal Naming Convention (UNC) format.

  Any of the following conditions can cause this command to fail:

  - The backup task cannot access a remote registry key.
  - <*source*> does not exist.
  - The backup task cannot write to <*destination*>.

- To back up the SMS site database, use

  sitedbdump <*source*> <*destination*>

  Where <*source*> is a database name and <*destination*> is the destination file that the backup task copies the database to.

- To back up server configuration information, use

  machinfo <*source*> <*destination*>

  Where <*source*> is the machine from which configuration data is backed up and <*destination*> is the destination file in which **machinfo** stores the collected configuration data in.

  The **machinfo** command uses Srvinfo.exe, Net.exe and IPConfig.exe to gather information SMS runs Srvinfo.exe from the SMS\Bin\i386 folder (this is where SMS Setup installs this executable file), and Net.exe and IPConfig.exe from the Windows system folder, so always ensure that the respective paths contains the latest version of these executable files.

  Important

  If the site server is configured with the Advanced Security mode and SQL Server is running on an operating system in the Windows Server 2003 family, then you need to manually gather configuration information from SQL Server. For more information about this issue, see article number 316360 in the Microsoft Knowledge Base at http://support.microsoft.com.

- To back up the site SQL Server configuration info, use

  smssqlinfo <*destination*>

  Where <*destination*> is the destination file name (without suffix) to which the backup task copies the SQL Server configuration data.

The **smssqlinfo** command requires that the Isql.exe tool be in the path. SMS Setup installs this tool to SMS\Bin\i386. However, several versions of Isql.exe can exist on the source computer. Ensure that the path accesses the latest version of this executable file.

Smssqlinfo generates the following three files:

- <*destination*>Data.txt
- <*destination*>Dboption.txt
- <*destination*>Helpdb.txt

Caution

Do not modify or delete the default commands in this section. If you do, you risk the completeness of the backup snapshot generated with this control file, and you might not be able to recover the site using it.

[Start]

Lists processes that you want restarted when the backup operation has been completed.

By default, SMS restarts all the services that were stopped. This includes the services that were stopped by default and all the services that you added in the **[Stop]** section using the **service** <*service name*> command.

The backup task does not attempt to start any services that are not installed, services that are already running, or processes that were stopped using the **exec** <*executable name*> command.

In this section, you can append the same commands as in the **[Stop]** section, with one difference: when starting executable files, you must enter the full path of the executable file, even if the executable file is in the current path.

The **sleep** command is useful when you need to start two processes, but one process cannot start until the second process is fully initialized. However, it takes some time for the first process to initialize. In this case, start the first process, and use the **sleep** command to pause the backup task until the first process is fully initialized. Then start the second process.

In case of failure of the backup task, it restarts only the three basic services:

- %*SITE_SERVER*%\SMS_SITE_COMPONENT_MANAGER
- %*SITE_SERVER*%\SMS_EXECUTIVE
- %*SITE_DB_SERVER*%\SMS_SQL_MONITOR

General Editing Guidelines

- If <*source*>, <*destination*>, or <*process name*> contains spaces or tabs, then they must be back quoted, as in: `process name with spaces`.

- To include a literal back quote, percent sign, tilde (~), or any other special character, you must precede it with a tilde. For example, to reference a token named "100% completed" type:

  %100~% completed%

- Use the following to include comments in the file:
  - To add a single comment line, include the # character at the beginning of a line. For example:

    ```
    # This is a single comment line.
    ```

  - To add multiple comment lines (a comment block) use the "#stop" and "#start" strings to delimit the beginning and the end of the comment block. For example:

    ```
    #stop
    This is the beginning of my comment block
    and this is the end of my comment block
    #start
    ```

    Comment blocks can be nested, but ensure that the section headers, such as [Tokens] or [Stop], are never commented out.
- You can specify <*source*> and <*destination*> paths with a UNC or drive letter format.
- Do not modify the section headings ([Tokens], [Stop], [Tasks], and [Start]), and do not modify their order.
- The backup control file, SMSbkup.ctl is not case sensitive.
- The backup task backs up each <*source*> only once. If a <*source*> token in a subsequent backup command resolves to a <*source*> that was already backed up by a previous backup command, then the subsequent backup command is ignored.
- If there are commands that operate on remote computers, you must ensure that the appropriate permissions to run these commands are in place. For example, to start a service on a remote machine, the SMSService account must have rights to start and shut down services on the remote machine.

File Name Conventions

The default SMSbkup.ctl file follows these naming conventions for the backup snapshot files:

- The prefix "SMSbk" is included in all file names to indicate that the files are part of an SMS backup snapshot.
- The type of server from which the data is being backed up is included in all file names.
  - Site = site server
  - Prov = site provider server
  - SQL = SMS site database server

- Each file name includes the data type.
  - Reg = registry keys
  - DB = database dump
  - ConfigNT = Windows NT® configuration data
  - ConfigSQL = SQL Server configuration data
- Each file name includes the actual data source. For example, “NAL” = NAL registry key.
- Extensions are based on the data type of each file:
  - .dat = registry keys and database dumps (i.e. raw data)
  - .txt = readable data

For example, the file with the backup of the NAL registry key from the site server is named SMSbkSiteRegNAL.dat

It is recommended that you follow the default naming conventions when you add backup commands to the default backup control file. Following these conventions allows you to easily identify the content of each file in the backup snapshot.

Important

If you modify SMSbkup.ctl file, closely monitor the next backup cycle. You might have introduced syntax errors in the file, in which case the task fails shortly after it starts. Fix these errors and rerun the task.

Using AfterBackup.bat to Archive a Backup Snapshot

To ensure that a recent backup snapshot is always available, it is recommended that you archive the backup snapshot every time the SMS backup task completes a backup cycle. Create several archives of the backup snapshot on removable media, and store it in different secure locations.

You can use the AfterBackup.bat file to run a third-party tool that automatically archives the backup snapshot every time you back up your site. After successfully backing up the site, the SMS backup task runs the AfterBackup.bat batch file. The AfterBackup.bat file integrates the archive and the backup operations, thus ensuring that every new backup snapshot is archived.

If your organization is using an archive solution that already archives data at the location that the Backup SMS Site Server task writes to, then you might choose to not create the AfterBackup.bat file. In this case, be sure to coordinate the SMS backup task schedule and your organization archive schedule as follows:

- Schedule the organization archive of the SMS backup snapshots at least as often as the Backup SMS Site Servers task runs.

- Schedule the organization archive of the SMS backup snapshots and the Backup SMS Site System task so they do not run at the same time. If both activities run at the same time, your organization archive might not copy the SMS backup files while they are being written, or the SMS backup task might not be able to write to the backup files while they are being archived.

Although the intended use of AfterBackup.bat is to archive SMS backup snapshots, you can use that file for other tasks that you need to perform at the end of every back up operation, such as:

- Run a SQL Server DBCC test to verify that there are no integrity problems with the SMS site database.
- Run a site health tool, or other health tools.

► To use the AfterBackup.bat file

1. Prepare an ASCII file with commands that archive your backup snapshot, or that perform any other post-backup tasks your site requires.
2. Name the file "AfterBackup.bat" and save it in the SMS\inboxes\smsbkup.box folder. Now, every time the backup task runs successfully, it will run the AfterBackup.bat file.
3. Every time after the AfterBackup.bat file archives the site's backup snapshot, store that archive in a secure location.

Scheduling Considerations for the Backup SMS Site Server Task

SMS is implemented in many different hierarchy structures, so it is not possible to have a single backup strategy that is appropriate for all hierarchies. Sites are different in their capacity, importance, and activity level. There is no single answer to the question, "How often should I back up my site?"

Obviously, when you back up data more often, there is less risk of losing data due to a site failure. However, backing up the site interferes with the site's regular operations and requires more resources. You must weigh the needs of each site and determine the backup frequency.

How Often Should You Back Up Your Site?

Use the following guidelines to help you decide how often to back up your site:

- Back up your site frequently enough to avoid a large difference between the data in the backup snapshot and the site data at the time of a site failure.

  For example, on large, busy sites where SMS tasks such as site configuration changes and advertising new programs are continually performed, back up the site daily.

  It might be adequate to back up a low-key site once or twice per week. However, if the SMS site database is small, the backup task is completed quickly. Because the backup task in this case has minimal effect on the site while it is running, consider backing up that site on a daily basis, too.

- Back up your site at a frequency that does not cause the site to fall behind on site activities. The frequency depends on the activity level of the site. For example, if the site is busy for almost twenty-four hours every working day, then you might not be able to schedule daily backup tasks. In this case, you might be able to back up the site only on days with less activity.
- Back up your site so it is cost effective to your organization. Weigh the cost of spending resources on backup versus the cost of spending resources to handle partial data loss after a site failure. This applies to all SMS site systems.

Note

The site server computer must be running when backup is scheduled to run. Otherwise, the backup task does not run and there is a big gap between backups. This is especially important if the site backups are not very frequent.

When Should You Back Up Your Site?

When you have decided how often to back up a site, you need to decide the appropriate time of day to perform the backup operation.

Use the following guidelines to help you make that decision:

- The backup task must run when the site server and SMS site database are not being used. While the task is running, no process should access the SMS site server or the SMS site database. To avoid the risk of corrupting data, no data should be processed at that time.
- The backup task must run at a time that allows post-backup activities to be completed before the site must continue its regular operation. Schedule the task to run at a time that allows the backup task to be completed, and that allows the site to process any backlog that accumulated while the back up task was running.
- The backup task must run at a time in which it does not conflict with other site activities. Tasks that take a long time to finish, such as database maintenance tasks, should not be interrupted, if possible. Schedule these tasks so they are completed before the Backup SMS Site Server task is scheduled to start.

Out-of-Schedule Backups (Event-driven Backups)

An out-of-schedule site backup might be necessary when performing certain operations on the site. For example, backing up the site server is a required step when swapping the site server computer. For the following site operations, backing up the site is not a required step, but it is important to back up the site before and after performing these operations.

Regardless of how you schedule the backup task, you should always have a recent backup snapshot before starting these operations. If the site's backup snapshot is not recent, then before starting these operations, back up the site and ensure that the backup task has been completed. After completing these operations, back up the site again.

Occasionally, you might need to perform a major upgrade to a system on which an SMS site is running. Backing up the site prior to the upgrade ensures that you can revert the system to its previous state, in case the upgrade fails. An out-of-schedule backup operation might result in a backup snapshot that is not archived. This can happen if the archive has a set schedule. In this case, you should manually archive the backup snapshot.

These operations include:

- Upgrading the site's database server or making configuration changes in SQL Server to items such as user connections, locks, and device allocation information (such as device size and type).
- Upgrading the operating system of the site server.
- Upgrading SMS. If you need to perform a series of site upgrades, it is important to back up the site before and after every upgrade.

Important

You cannot restore an SMS 2.0 backup snapshot to an SMS 2003 site. Use the SMS 2.0 backup snapshot only if the upgrade failed, and you plan to revert the system back to SMS 2.0.

- Modifying the site hierarchy.
- Running site reset.
- Upgrading the site from Standard Security to Advanced Security.
- Modifying the site's accounts.

After you schedule the backup task in the SMS Administrator console, it can take up to 24 hours for the schedule change to take effect. In this situation, you can initiate a manual backup cycle by starting the SMS_SITE_BACKUP service on the site server.

Start the service to initiate a manual backup cycle after running SMS Setup, or to run an unscheduled backup when performing important configuration changes to the site. Such manual backup has no effect on the schedule of the backup task.

Important

A manual backup cycle fails if **Backup destination** is not set in the **Backup SMS Site Server Properties** dialog box. You can set this entry even if the backup task is disabled for the site.

You can start the SMS_SITE_BACKUP service remotely by using the operating system services tool, or by using the SMS Service Manager if you have administrator-level access to the server.

Enabling and Configuring the Backup SMS Site Server Task

By default, the Backup SMS Site Server task is disabled. To start backing up your site, you must enable and configure the task. You need to specify values, such as how often you want the backup task to run, and where to store the backup snapshot.

► To enable and configure the SMS backup task

1. Navigate to **Tasks** in the SMS Administrator console.

```
Systems Management Server
 └ Site Database
  └ Site Hierarchy
   └ <site code>-Microsoft
    └ Site Settings
     └ Site Maintenance
      └ Tasks
```

2. In the right pane, double-click **Backup SMS Site Server**.
3. In the **Backup SMS Site Server Properties** dialog box, enable the backup task, and then specify the following:

   Schedule

   Specify the schedule for the backup task. Specify the day of the week and the time period for the task to start. The time period is the time interval in which the task can start, and it is defined by **Start after** and **Latest start time**.

   Note

   The time period must be at least one hour long, and it must start and end within a single day.

   The schedule that you specify is a recurring schedule. The task runs every week, on the day and in the time period that you specify.

   Backup destination

   Enter a folder in which the backup task stores the backup snapshot. You can specify a local path or a remote path for the backup destination. In either case, the backup destination must exist, or be such that SMS can create it.

   The backup destination must be in one of the following formats:

   - A UNC path, such as \\Central\BackupSMS
   - A drive letter path, such as F:\\BackupSMS

   You cannot specify the backup destination to be within the SMS folder. Also, you must ensure that the backup destination folder has sufficient disk space and is secure, so that the site data stored at that location is protected.
4. Use the SMS Service Manager to enable logging for the Backup SMS Site Server task.

Verifying Success of the Backup SMS Site Server Task

After the Backup SMS Site Server task is completed, verify that it was completed successfully. When it is successful, the backup snapshot that was generated is valid, and you can successfully use it if recovery becomes necessary.

Verify that a backup snapshot is valid by:

- Verifying that the time stamp of the backup destination folder, and the files in that folder, is the time that the backup task ran.
- Verifying that there are no errors logged in the SMSbkup.log file.
- Checking status messages for backup. The success status message reads, "Backup completed successfully."
- Running DBCC checks from SQL Server after backup has been completed to verify that the database is intact. The assumption is that if the database is valid before the backup and after the backup, then the backup snapshot is most likely valid.
- Performing an occasional live recovery using the latest backup snapshot of the site. For more information about live recovery test, see Chapter 13, "Maintaining and Monitoring SMS Systems."

Backing Up a Secondary Site

Secondary sites can be as significant as primary sites. Because secondary sites can also fail, you need to back them up. You back up a secondary site in the same manner that you back up a primary site, by using the Backup SMS Site Server task.

The main difference between backing up a secondary site and a primary site is the lack of an SMS site database at the secondary site.

When you back up a secondary site, the following differences apply:

- When you perform the steps in the "Backing Up a Site Using the Backup SMS Site Server Task" section, ignore the step that verifies database integrity.
- The backup task ignores the following entries in the backup control file when backing up a secondary site:
  - Database related commands: **smssqlinfo** and **sitedbdump**.
  - All backup commands in the [Tasks] section, in which the <*source*> is a database server.

  If you customize the backup control file, you do not need to remove the unnecessary database-related commands. When the backup task runs on a secondary site, it ignores these entries.
- When the backup task runs on a secondary site, it does not stop the SMS_SQL_MONITOR service because it does not exist on a secondary site.

Backing Up the Central Site

Backing up the central site is unique in the following ways:

- It is the busiest site in the hierarchy, leaving less time available to back it up.
- It has the largest amount of data that needs to be backed up, which increases the length of the backup operation.

As a result, you might attempt to reduce the risk of backlogs resulting from backup operations on the central site by planning to back up your central site less often. Because the central site is the most critical site in the hierarchy, this is not recommended.

Recovering the central site is unique in the following ways:

- On the central site, many configuration changes can be constantly made, and it might be impossible to repeat all of them.
- The central site does not have a parent site from which to obtain a copy of the site control file.

To resolve these issues, and to ensure that it is possible to recover all the data of the central site, follow these recommendations:

- Back up the central site frequently.
- Have a reference site for the central site, so that recovery tools can recover packages and advertisements created after the site is backed up.
- Have a copy of the that is as recent as possible.
    - Between site backup cycles, back up the central site control file frequently. You can use system tools or batch files to automatically back up the site control file frequently. To back up site control files using the Hierarchy Maintenance tool (PreInst), see the "Recovery and Repair Tools" section later in this chapter.
    - Store the site control file backup at the designated reference site.
    - Store the site control file with a *.CT0 format so that during recovery it is sufficient to rename the file, and drop it on the recovering central site.

If you follow these recommendations, then you will be able to recover most of the configuration when recovering the central site. You will be able to recover configuration data and software distribution objects definition from the reference site. The administrative data that is lost if the central site fails is less critical, and is regenerated after the central site regains functionality.

Monitoring Backup

The Backup SMS Site Server task logs its activity to the SMSbkup.log file located in the SMS/Logs/ folder.

The task also generates status messages to report its activity and to report various erroneous conditions encountered at run time. (Backup status message numbers are 50xx.) There are three levels of errors reported by the SMS backup task: Critical Errors, Errors, and Warnings. Error conditions affect the backup task, depending on the error level.

Critical errors Critical Errors cause the backup task to stop. The backup operation is considered failed.

Errors The backup task continues to run when encountering Error conditions; however, the backup operation is considered failed and the snapshot (if generated) is not valid.

Warning The backup task continues to run when encountering Warning conditions. A warning by itself does not invalidate the backup snapshot and does not cause any harm to the backup operation. The backup operation is considered successful.

Note

If AfterBackup.bat runs as part of the backup operation, it has no effect on the success or failure status of the backup operation.

The backup task logs Critical Errors when encountering problems such as:

- The backup task is unable to read the registry to obtain site setup information.
- The backup task is unable to access the site control file, or the backup control file (SMSbkup.ctl).
- The specified backup snapshot destination is not valid (for example, it points to the SMS installation folder).
- The backup control file has syntax errors.

The backup task logs Errors when encountering problems such as:

- The backup task cannot perform a registry backup command (**reg**) or a database backup command (**sitedbdump**).
- The backup task cannot start a service.

The backup task logs Warnings when encountering problems such as:

- The backup task is unable to connect to the system's Service Control Manager to verify that a service is installed.
- A token is being redefined in the SMSbkup.ctl file. In this case, the second definition of the same token is ignored.
- The backup task cannot perform a file backup command (**file**).
- Some of the tools in **machineinfo**, or some of the commands in **smssqlinfo**, failed.

If the backup task reports any Critical Errors or Errors, find the cause of the problem, repair it, and rerun the task.

Using Third-Party Backup Tools to Back Up SMS Sites

You can use a third-party backup tool, or you can write a custom script to back up your site. However, to successfully back up your site using such non-SMS backup tools, they must interoperate with SMS recovery functions by complying with the following guidelines.

To produce a valid backup snapshot, any custom tool or script must:

- Carry out the exact same tasks in the exact same order as they appear in the default SMS backup control file (SMSbkup.ctl).
- Back up the site as a snapshot of all data, at a time when the SMS site database is not actively accessed. Stop all processes that might access the site's database, such as SMS services.

  If processes access data while it is being backed up, partially completed tasks can be in the backup. As a result, the data in the registry, the files, and the database might not be synchronized with each other. This can lead to problems after a site recovery.
- Produce a backup snapshot, identical in its directory structure and file names to the one produced by the default SMS backup task.
- Include, at a minimum, all the data produced when using the default SMS backup control file, SMSbkup.ctl.

The simplest and most reliable plan actually allows you to benefit from both the SMS backup task and a third-party archive tool. Use the SMS backup task to create a backup snapshot, and use a third-party tool to archive the backup snapshot to a secure location.

Recovering a Site

A site can experience different problems. Some problems might be easy to repair, and some problems might be more serious, requiring a total recovery operation to regain the site's functionality. This section describes the SMS site recovery operation. Before you decide to perform a site recovery operation, you must troubleshoot the site and determine whether a recovery operation is the appropriate remedy.

Important

Due to the complexity of the procedures involved in recovery, the accuracy these procedures require to be carried out, and the critical importance of a successful recovery, information about recovery is intended to be used by experienced SMS administrators, Microsoft Consulting Services consultants, solution providers and, technical support engineers who have a deep technical knowledge of SMS and the environment it is being used in.

Determining Whether a Site Recovery Operation Is Necessary

An SMS site may be exhibiting various failure symptoms. The severity of the failure symptoms is not always a good indicator of the severity of the underlying problem. It is important to correctly diagnose the problems that the site is experiencing, and then to recover the site if appropriate. This section provides some guidelines that can help you determine whether a recovery operation is necessary.

SMS stores most of the site data in the registry, in system files, and in SQL Server databases. To function properly, data integrity and synchronization, among all data stores in SMS, is an absolute necessity. Therefore, even if only one data store is corrupted, to prevent data corruption, a snapshot of all data stores must be restored. Restoring all of the site's data must be performed as part of a recovery operation.

Based on the data integrity requirement of SMS, if your site is failing due to any of the following reasons, you must perform a recovery operation:

- The computer that the site server or the SMS site database server is running on has a failing operating system.
- The drive that the operating system, SQL Server, or SMS is installed on is failing.
- The file system of the site server has become corrupted.
- SQL Server is failing and it must be restored.
- The SMS site database has become corrupted.

Supported Configurations and Recovery Scenarios

You can recover a site with a recent or an old backup snapshot, or even without a backup snapshot. The amount of data restored depends on whether there is a recent site backup snapshot and whether SMS can obtain data from other sites in the hierarchy.

When recovering a site without a recent backup snapshot, expect a significant data loss. Without a recent backup snapshot, the SMS site database is not restored, and all inventory, status messages, and object definition data is lost. Using a recent backup snapshot dramatically increases the amount of data recovered, although some data loss is still expected.

If you are planning to recover a site using a recent backup snapshot, then SMS supports restoring an SMS site from a backup snapshot only if:

- The backup snapshot is restored to the original site that was backed up.
- The version and service pack of SMS are the same for both the original site and the restored site.
- The version and service pack of SQL Server are the same for both the original site and the restored site.
- The recovered site name is identical to the site's original name.

- The recovered site belongs to the same domain as the original site.
- No site accounts were changed between backup and restore.

The operating system of the original site and the recovered site can be different. However, it is not recommended that the operating system of the recovered site be of an earlier version than the operating system of the original site. When restoring data from the backup snapshot, SMS restores registry keys, files, and database. The operating system version has no effect.

If you plan to recover a site without restoring a backup snapshot, then SMS supports:

- Recovering the site with the same SMS service pack as the failed site had, or with a later SMS service pack.
- Recovering the site with the same version or service pack of SQL Server as the failed site had, or with a later version or service pack of SQL Server.
- Recovering the site with the same Microsoft Windows NT service pack as the failed site had, or with a later Windows NT service pack.

SMS does not support the following recovery scenarios:

- Restoring a backup snapshot, that was created before an SMS upgrade, to the upgraded site.
- SMS site systems installed on Windows 2000 servers running Terminal Services.
- SMS site systems installed on Windows 2000 servers running Terminal Services Client.

The Recovery Procedure

To successfully recover a site, you must prepare for the recovery operation and perform post-recovery tasks. The following procedure describes the entire recovery operations.

► To recover a site

1. Notify other SMS administrators and SMS users about the site failure.
2. Prepare for the recovery operation. For information about preparing for a site recovery operation, see the "Preparing for a Site Recovery Operation" section later in this chapter.
3. Disable the backup task. If the site is accessible, ensure that the backup task does not run during the recovery operation. If it runs, it will interfere with the recovery operation.
4. Designate reference sites that the Site Repair Wizard can use. For information about designating reference sites, see the "SMS Site Repair Wizard" section later in this chapter.
5. Run the Recovery Expert from the Recovery Expert Web site and print the site recovery task list.
6. Recover the site by performing all the tasks prescribed by the Recovery Expert, in the order that they are listed. Use other recovery tools as indicated by the recovery tasks. For more information about recovery tools, see the "Recovery and Repair Tools" section later in this chapter.

7. Verify that the site is successfully recovered by following the respective Recovery Expert tasks. Do not attach any child sites to a recovered site until you are certain that the site was successfully recovered.
8. Restore all custom files that were manually backed up, such as custom SMS Administrator console files (.msc files), custom MOF files (such as SMS_def.mof), and Supplemental Reports.
9. If the software update management feature is used, then restore the Definitive Software Library to its original folder. For information about restoring software update management packages without the Definitive Software Library or without the related objects, see the "Managing the Site After Recovery" section later in this chapter.
10. Investigate the cause of the site failure, and make any necessary adjustments to ensure that this failure will not repeat.
11. Schedule recurring backups on the recovered site.

The remainder of this chapter further describes site recovery. Before starting the recovery operation, read the rest of the topics in this section.

Recovery and Repair Tools

To assist you in a successful site recovery operation, SMS provides various recovery and repair tools, the main one being the Recovery Expert. These tools are automatically installed with SMS, with the exception of the Recovery Expert, which administrators must set up before they can use it.

During a site recovery operation, recovery and repair tools greatly simplify some recovery tasks, reduce the risk associated with editing low-level data, and perform tasks that are impossible to perform by using any other method.

Caution

Failing to use recovery tools appropriately can significantly interrupt site operations, or cause unrecoverable loss of data.

The recovery and repair tools set includes the following tools:

Recovery Expert Guides you through the recovery process by generating a recovery task list based on the site's specific failure scenario and site configuration.

SMS Site Repair Wizard Automates some of the Recovery Expert's tasks, and helps recover some of the data that was not backed up. Using the SMS Site Repair Wizard eliminates user errors that might occur when performing complex tasks.

ACL Reset (ACLreset.exe)
Resets access control lists used by the SMS Server Connection account and remote site systems to access the site server.

Hierarchy Maintenance tool (PreInst) Passes commands such as site repair or site diagnostics commands to the SMS Hierarchy Manager while the SMS Hierarchy Manager is running.

Unenforce Software Metering tool (Unenforce.exe) Overrides software metering enforcement rules. This utility is needed only when recovering an SMS 2.0 site, and it is included on the SMS 2003 CD for compatibility reasons. For more information about the Unenforce Software Metering tool, see the SMS 2003 Help.

The Recovery Expert

The Recovery Expert is a Web-based recovery tool that guides you through a site recovery operation. When you run the Recovery Expert, it scrolls through a series of Web pages with questions about the site failure scenario and the site configuration. The Recovery Expert then evaluates your answers and presents a recovery task list. Perform these tasks, in the order that they are prescribed, to recover the failed site.

Recovery tasks vary from site to site and from one failure scenario to another, but most recovery scenarios consist of the following phases:

1. Rebuilding the failed servers.
2. Restoring the site data.
3. Repairing and re-synchronizing data. These are the core tasks of a site recovery, and they are required to prevent interruption of operations and corruption of data.
4. Verifying the success of the recovery by testing the functionality of the recovered site.

Each recovery task belongs to one of these phases. A recovery tasks list produced by the Recovery Expert typically contains tasks from all phases.

Running the Recovery Expert

To use the Recovery Expert, you must first set up a Recovery Expert Web site. For information about how to set up a Recovery Expert Web site and how to run the Recovery Expert, see the "Setting Up a Recovery Expert Web Site and Running the Recovery Expert" section earlier in this chapter.

During site recovery, you run the Recovery Expert, and then you perform all the recovery tasks as prescribed. Recovery tasks indicate when you can use the SMS Site Repair Wizard tool to automate recovery tasks, and when you should use other recovery tools.

SMS Site Repair Wizard

The SMS Site Repair Wizard automates complicated recovery tasks and tasks that would be impossible to perform otherwise. Using this tool simplifies site recovery, increases the amount of data recovered, saves time, and reduces the risks associated with recovery.

The SMS Site Repair Wizard is used in conjunction with the Recovery Expert during a site recovery operation. Using the SMS Site Repair Wizard during site recovery is strongly recommended. Each recovery task in the Recovery Expert contains a note stating whether it can be automated by using the SMS Site Repair Wizard.

Depending on the site backup schedule and on the activity at the site, the latest site backup snapshot might not include the most recent modifications to the site. Any changes made after the most recent site backup are not included in the site's backup snapshot. As a result, after restoring the site backup snapshot, the site can be out of synchronization with the rest of the hierarchy. For example, the site's backup snapshot might contain information about a child site that has since moved to a different parent site.

After restoring the site backup snapshot, the SMS Site Repair Wizard attempts to restore as much as possible of the data that was not backed up. The wizard can restore objects such as collections based on query rules, packages, programs and advertisements, but cannot restore data such as software metering rules, reports, and custom queries. The SMS Site Repair Wizard restores data by restoring site settings and synchronizing site objects with parent and child sites.

Important

Running the SMS Site Repair Wizard independently is not recommended. Always run the Recovery Expert first, and then run the SMS Site Repair Wizard as directed by the Recovery Expert.

Restoring site settings When running the SMS Site Repair Wizard, the user is prompted to enter any changes to site settings that occurred after the most recent site backup. The wizard then restores site settings according to the user input. For example, the administrator can specify that a child site no longer reports to the recovering site. The SMS Site Repair Wizard then deletes all objects associated with that child site from the recovering site.

To restore site settings, the wizard also uses the parent site, if exists. The wizard obtains the most recent copy of the recovering site's site control file. It then uses this file to configure the recovering site.

Synchronizing objects The SMS Site Repair Wizard synchronizes objects between the recovering site and other sites in the hierarchy, as follows:

- The wizard restores control to objects, such as collections based on query rules, packages, programs, and advertisements, that were created on the failing site after the latest site backup was completed, but before the site failed. After restoring the site's backup snapshot, the recovering site does not contain those objects because they are missing from the site's backup snapshot.

  Objects are regularly replicated from one site to other sites in the SMS hierarchy. This allows the wizard to use designated reference sites to replicate these objects from other sites to a recovering site. After these objects are restored to the recovering site, the recovering site has full control over these objects and they are synchronized between lower sites in the hierarchy and the recovering site. For more information about designating reference sites, see the "Designating reference sites" section.

- The wizard deletes objects at the recovering site that were inherited from upper level sites, but were then deleted at the originating site. Objects that were created at upper level sites might have been deleted, while the site's most recent backup snapshot still contains them. After restoring the site's backup snapshot, the recovering site contains these objects. The wizard checks all inherited objects that exist on the recovering site. It then checks if these objects exist at the parent site. Inherited objects that exist on the recovering site, but no longer exist on the parent site, are deleted.

Designating reference sites

The recovering site can regain control of orphaned objects only if you designate reference sites. Reference sites are lower level primary sites to which objects from a failing site were replicated. The SMS Site Repair Wizard reads object definitions at the designated reference sites, and then uses those definitions to recreate the missing objects at the recovering site.

If the failing site is the only site in the hierarchy, or if all other sites in the hierarchy are not functioning properly, then you cannot designate reference sites, and you will not be able to recover a significant amount of data.

When you run the wizard, it prompts you for reference sites designations. When designating reference sites, you should consider the following:

- A reference site must be a primary site.
- The number of objects that can be replicated down the hierarchy before a site failure depends on network speed and timing. A reliable, high quality network connection between the recovering site and its reference site ensures that:
  - Definitions are replicated quickly. If a site fails, chances are higher that objects created just prior to failure were replicated to lower level sites.
  - During a recovery operation, the wizard can quickly obtain object definitions from the reference site.
- Multiple reference sites may increase the amount of objects recovered. Designate more than one reference site as follows:
  - 1-2 reference sites with high quality, reliable connection are sufficient
  - 3-5 reference sites if the connections are not of the highest quality
  - Designate reference sites from different tiers in the SMS hierarchy
- Having a reference site at a close physical location to the recovering site is helpful.

Running the SMS Site Repair Wizard

Before running the SMS Site Repair wizard, you must ensure that there are no open Administrator console windows. Ensure that the current user has at least **Read** permission to objects such as collections, packages and programs on the designated references sites, and on the parent site, and administrative credentials on the recovering site.

When you run the Recovery Expert, it prompts you whether you intend to use the SMS Site Repair Wizard. If you chose to use the wizard, then the Recovery Expert produces the recovery task list, with the following differences:

- All tasks that can be automated by the SMS Site Repair Wizard are unavailable.
- The task list contains the **Run the SMS Site Repair Wizard** task.

As you start to perform the recovery tasks in the order prescribed by the Recovery Expert, do not perform the tasks that are unavailable. When you reach and run the **Run the SMS Site Repair Wizard** task, the wizard completes all the tasks that are unavailable. When the wizard finishes, continue to perform the remaining tasks in the list.

Note

All tasks that can be automated by using the wizard are treated as a set. When the wizard runs, it performs all the tasks in that set.

The SMS Site Repair Wizard operates in two stages. During the first stage, the wizard restores the site backup snapshot to the recovering site. During the second stage, the wizard determines what modifications were not included in the site backup snapshot, and attempts to reapply as many of these modifications as possible.

The wizard logs its activity to c:\SMS\Logs\sms_srw.log.

Note

Depending on the size of the database, it might take a considerable about of time for the wizard to restore it. As soon as the wizard submits the database restore SQL command, it logs a message stating that the database restore operation has started. If the wizard seems inactive, check the log file. The wizard might be busy restoring a large database.

ACL Reset Tool

ACL Reset is a command-line tool that resets the access control lists used by the SMS Server Connection account and by remote site systems to access the site server. ACL Reset does not reset access permissions to non-SMS objects. You can find the ACL Rest tool (ACLreset.exe) in the SMS\bin\1386\<*language*> folder.

ACL Reset is a repair tool and an important recovery tool. During a recovery operation, Recovery Expert tasks direct you to use this tool primarily to perform the following tasks:

- Create a new SMS Server Connection account, even if it is recreated with the same name. This ensures that SMS processes that rely on the SMS Server Connection account have the correct permissions to objects on the site server.
- Restore the SMS or NAL registry key, or any subkeys under them.
- Restore the SMS folder tree, or any files or subdirectories under it.

You also must use ACL Reset when performing operations such as:

- Changing the SMS Server Connection Account.
- Resetting the SMS Server Connection Account.

For more information about using ACL Reset and ACL Reset syntax, see the SMS Help.

Hierarchy Maintenance Tool

The Hierarchy Maintenance tool passes commands to the site's Hierarchy Manager while the Hierarchy Manager is running. You can use the Hierarchy Maintenance tool to diagnose problems in a site, to repair sites, to dump site control images, to distribute public keys, or to stop all SMS services at a site. You can find the Hierarchy Maintenance tool (PreInst.exe) in the SMS\bin\1386\<*language*> folder.

The Hierarchy Maintenance tool requires that you enable the Hardware Inventory Client Agent for the SMS site. This allows the client to determine whether or not particular servers are local to the client.

The Hierarchy Maintenance tool is both a repair tool and an important recovery tool. During a recovery operation, Recovery Expert directs you to use this tool primarily to: perform the following tasks:

- Update package definitions at the recovering site with updates from the originating site, if the recovering site has packages inherited from upper level sites. After updating the recovering site, these changes propagate to lower level sites.
- Restore any changes to feature configuration that were not backed up. Feature configurations are stored at the site control file, which is then forwarded to the parent site. You can use the Hierarchy Maintenance tool to dump the site control file from the parent site to the recovering site, so that the recovering site to be reconfigured the way it was.
- Restore objects that were created after backup and that were lost when the site failed.
- Dump site control files.
- Securely exchange public keys between the recovering site, its parent and its child sites.

For more information about using Hierarchy Maintenance tool, see the SMS Help.

Preparing for a Site Recovery Operation

Before you can start recovering the site, you must prepare as follows:

- Analyze data traffic issues and reduce the traffic load as much as possible. This includes intrasite data traffic and site-to-site data traffic. For more information, see the "Data Traffic Issues" section later in this chapter.
- Analyze security issues. For more information, see the "Security Issues" section later in this chapter.

- If you have been regularly backing up the site, then ensure that:
  - You can access the most recent backup snapshot.
  - The log file of the most recent site backup operation indicates that the site was backed up successfully.
  - If any integrity tests were performed to ensure the integrity of the site's backup snapshot (such as DBCC test), log files indicate that these tests have passed successfully.
  - The site backup task is not scheduled to run during the recovery operation.
- Obtain the most recent copy of the hierarchy configuration document.
- Ensure that the Recovery Expert Web site is set up and that you can connect to that site and run the Recovery Expert. For information about how to set up and run the Recovery Expert, see the "Setting Up a Recovery Expert Web Site and Running the Recovery Expert" section earlier in this chapter.
- Ensure that you can run the rest of the recovery and repair tools from the failing site, or from a remote server.

Data Traffic Issues

While the failing site cannot maintain its regular communication with the rest of the hierarchy, the site clients and other sites in the hierarchy continue to generate data such as discovery records, inventory data, and status messages. While the failing site is disconnected, backlogs of pending data are queued on various client and server systems waiting for the next opportunity to transmit. The size of the data backlog depends on many factors, but the critical factor that affects all types of traffic is how long the site is out of service.

As soon as the site server is reconnected to the rest of the hierarchy, the backlog of data is processed and, if appropriate, forwarded up the hierarchy. If the site server was offline for a long time, a large amount of data might be suddenly overflowing the network, when it all tries to reach the site server. The site server might not have the capacity to handle such an extreme amount of data. That is why it is important to minimize the backlog as much as possible and to complete the recovery operation as quickly as possible. But it is important to perform it carefully and precisely to ensure a successful operation.

The data traffic that you need to be aware of is intrasite traffic (traffic within the site, between clients and site systems, and between site systems and the site server), and traffic between sites. Intrasite traffic consists primarily of inventory data, discovery records and client status messages. Traffic between sites includes object definitions such as package and collection definitions, site status messages, and site configuration changes.

Intrasite data traffic

Clients continue to generate hardware and software inventory data, discovery data records and status messages. This data is usually forwarded to CAPs or management points, and then to the site server. Even while the site server is offline, clients forward this data to CAPs and management points. When the site server regains functionality, the data is forwarded from the CAPs and management points to the site server to be processed. This can flood the site systems with objects that must be processed all at once.

Intra-site data traffic can originate from the site's clients, from the site's CAPs or management points, or from child sites. The traffic load depends on the number of clients at the failing site and at lower sites and on other factors as follows:

- Data Discovery Records (DDR) data
  - Whether any discovery methods are enabled at the failing site or at lower sites
  - At sites where discovery methods are enabled — the scheduling of discovery methods
  - At sites where logon discovery is enabled — the frequency of user logon events
- Hardware and software inventory data
  - Whether hardware or software inventory are enabled at the failing site or at lower sites
  - At sites where hardware inventory or software inventory are enabled — the frequency of inventory
  - At sites where software inventory is enabled — whether file collection is enabled
  - At sites where file collection is enabled — the number and the size of files being collected
- Status messages
  - At sites where software distribution is enabled — the number and the frequency of advertisements that run on the client computers
  - At sites where software inventory or hardware inventory are enabled — the frequency of inventory

Site-to-site traffic

Sites will accumulate large amounts of data and transactions that need to be forwarded up and down the hierarchy, and then transmit when the inbound share becomes available to receive files. This traffic includes some client traffic such as DDRs that are queued for transmission up the hierarchy.

The amount of site-to-site traffic depends on the number of sites below the failing site, the number of clients in those sites, and how long the failing site is offline. It also depends on the following:

- Site control changes
  - Number and frequency of changes made to the site configurations at sites above or below the failing site
  - Number of sites below the failing site
- Packages and advertisements
  - The amount of software distribution related activity at the parent of the failing site, such as creating packages and advertisements, making modifications to packages, and refresh activity.
  - The number of packages and advertisements at the parent of the failing site
  - The size of the packages being distributed down from sites above the failing site

- Collections
  - The number of custom collections created at sites above the failing site
  - The frequency of collection property changes at sites above the failing site
- Status Messages
  - The number of sites and clients in the hierarchy below the failing site
  - The configuration of status message replication rules at the failing site and at lower sites
  - The amount of inventory and software distribution activity at the failing site and at lower sites

Reducing the traffic load

Much of the data generated while the failing site is offline will no longer be relevant after recovering the failing site. For example, status messages reporting the inability to connect to the offline site will be irrelevant after recovering the site. Other data such as hardware and software inventory will be gathered again after the recovery operation is completed. It is recommended that you minimize the data accumulated while recovering the failing site by doing any of the followings:

- Configure the status system:
  - At the failing site's child sites — configure the status system not to replicate status messages to the parent site.
  - Configure status message rules at lower level sites to discard as many status messages as possible.

  For more information about status filter rules, see Chapter 14, "Using the SMS Status System."
- Adjust the software and hardware inventory interval at lower level sites, so that inventory is not collected until the recovery operation is completed. Depending on the inventory schedule and on how long the recovery operation takes, this might help reduce the traffic. You need to consider the current inventory interval and the time it takes for the schedule change to take effect.
- Use the Hierarchy Maintenance tool (PreInst.exe) to remove pending jobs entirely. For more information about the Hierarchy Maintenance tool, see the SMS Help.

Security Issues

There are some security issues involved in a recovery operation if all of the following conditions exist:

- The site is part of a hierarchy.
- The site is configured with advanced security.
- There is no domain controller.
- The site recovery involves reinstallation of the operating system.

When reinstalling the operating system, the site's original private and public keys are lost, and new keys are generated. After recovering the site, the site will have a new public and private key pair. The new public and private keys will no longer match the original public and private keys that the other sites have for the recovered site. In this scenario, there is no domain controller to resynchronize the keys. The mismatched keys prevent communication between the recovered site and its parent, and between the recovered site and its child sites.

When using the Recovery Expert in this recovery scenario, the recovery task list includes a task that resolves this issue. It instructs users to run the Hierarchy Maintenance tool (PreInst.exe) with the appropriate parameters to generate text files with information about the new keys. The user is then instructed to manually copy these files to their appropriate location at parent and child sites.

For more information about security, see Chapter 5, "Understanding SMS Security," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide.*.

Managing the Site After Recovery

After you have successfully recovered the site, there might be issues with large amounts of data that accumulated while the site was offline. Also, there might be data that could not be recovered during the recovery operation that you might still be able to manually recover.

Estimating the Amount of Pending Data

After the site is fully recovered, even if you attempted to reduce the amount of data that accumulates, the amount of data waiting to be processed might still be more than the site's usual amount of data.

It might be helpful to get an estimate of the amount of pending data on the recovered site. On a parent or child site server, check the SMS\inboxes\schedule.box folder for files with a .job extension. These files represent pending jobs to send data of various types, including packages, site control changes, and status messages. A large number of files in this folder represent a large amount of pending data. Some of this data is contained in subfolders of the SMS\inboxes\schedule.box, and other data, such as package replication, transfers packages from the compressed package folder \SMSPKG, which might be much larger.

Mitigating Data Loss

Even in a successful recovery operation, some site data might be lost. When the site is fully functional after the recovery operation is completed, you can mitigate some data loss by updating the site configuration, regenerating inventory, and redistributing packages. Although you can mitigate most of the data loss after a site failure, it is better to invest efforts in providing valid, recent backups.

Recovering collections with direct membership rules

The Site Repair Wizard does not properly restore collections with direct membership rules that did not exist in the site backup snapshot. After the wizard restores those collections, they contain data that is not valid. After recovering the site, you can mitigate this by doing the following:

1. Delete the collection that is not valid on the recovered site and all child sites to which the collection has propagated.

2. Recreate the collection.
3. If any programs were advertised to this collection, delete the advertisements to the collection that is not valid, and advertise again the programs to the new collection.

Recovering software update management packages

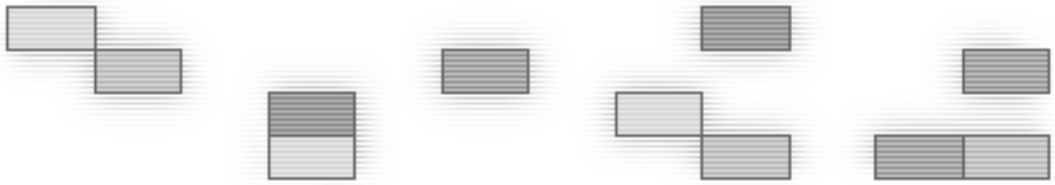
Software update management-related objects, such as package and advertisement objects, are restored during the site recovery operation. However, if the Definitive Software Library was not backed up, then these objects are useless and you must remove them. After removing those objects, you can use the Distribute Software Updates Wizard to recreate those packages.

If a backup of the Definitive Software Library exists, but for some reason the related objects are missing, you can restore the Definitive Software Library and then recreate those objects. Those objects might be missing if, for example, they cannot be recovered during a site recovery operation, or if they are accidentally deleted.

► To recreate package objects for existing software updates management package source files

1. If necessary, restore the Definitive Software Library to its original folder.
2. Create a new software update management package.
3. Import the XML file from the original package source folder.
4. Specify the folder name of the package source files so it is identical to the original folder name.

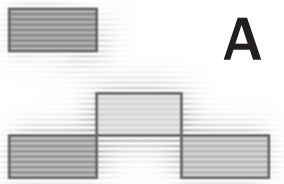
Appendices



The appendices of the *Microsoft Systems Management Server 2003 Operations Guide* contain useful information that will help you gain a greater understanding of Systems Management Server 2003 operations.

APPENDIX A

Using SMS to Distribute Office



Microsoft® Office XP is a suite of business products that includes word processing, spreadsheets, presentation software, and other programs that are useful in a business setting. Microsoft Systems Management Server (SMS) 2003 is an enterprise management tool that you can use to deploy and maintain Office XP in your organization. There is a question and answer section for deployment details and related material at the end of this appendix.

This appendix supplements the Office XP documentation and the SMS 2003 software distribution documentation by providing information specifically targeted at deploying Office XP to SMS 2003 clients in your organization. For more information about Office XP features, see the *Microsoft Office XP Resource Kit* at http://www.microsoft.com/office/ork/xp/default.htm.

In This Appendix

- Overview of Office XP Deployment
- Important Concepts and Issues
- Deploying Office XP in an Enterprise
- Maintaining and Updating Your Office XP Installation
- Using Resilient Sources
- Frequently Asked Questions

Overview of Office XP Deployment

This section describes the advantages of and requirements for using SMS 2003 to deploy Office XP.

Advantages

SMS can be used to reduce the cost and the time required to deploy Office XP. Some reasons you might want to use SMS 2003 to deploy Office XP in your organization are:

- To scale successful deployments from your test lab, to pilot users, through verification, and throughout your enterprise.
- To deliver and complete the installation during off hours to reduce the effect on your users.
- To coordinate upgrades across multiple sites.
- To manage slow links between sites.
  - SMS 2003 provides administrators with the ability to compress package sizes, which minimizes the bandwidth usage.
  - SMS 2003 allows administrators to throttle and schedule bandwidth usage so that SMS reduces its concurrent network usage.
- To deploy Office XP to a large number of clients.
- To force the installation of Office XP on client computers.
- To take advantage of status reporting and troubleshooting tools.
- To deploy Office XP to different types of clients that are running Microsoft Windows® operating systems, such as Microsoft Windows XP Professional, Windows 98, Windows NT® Workstation 4.0, and Windows 2000 Professional.

Other advantages of using SMS 2003 are:

- SMS administrators can verify a pilot deployment of Office XP, and then expand the collection targeted with Office XP to effectively scale a deployment that meets corporate requirements.
- You can use the SMS administrative context to deploy Office XP to computers where users do not have administrative credentials on the local computer.
- You can use SMS reporting through the status message system on delivery and execution.
- You can create SMS collections in order to install Office XP only to computers that meet your corporate standards, and then install to the remaining computers when they meet the requirements. For example, a corporation might determine that only systems with at least 64 MB of RAM should upgrade to Office XP. As RAM is upgraded, these computers become part of the dynamic collection of computers with 64 MB of RAM, and then they install Office XP.

Assumptions

This appendix is targeted at SMS administrators. Before you attempt to use SMS 2003 to deploy Office XP, verify that:

- You are familiar with Office XP and SMS 2003.
- You are familiar with SMS 2003 administration and software deployment. If not, see the SMS 2003 documentation.
- SMS 2003 or SMS 2.0 SP2 or later is deployed in your organization. If you are supporting clients running Windows XP, SMS 2.0 SP4 or SMS 2003 is recommended. Microsoft does not support the use of SMS version 1.2 to install Office XP. For information about upgrading SMS, refer to the Systems Management Server Web site at http://www.microsoft.com/smserver/default.asp.
- You are familiar with the *Microsoft Office XP Resource Kit*. The administrator should be very familiar with the Office Profile Wizard and the Custom Installation Wizard in Office XP. These topics and others are covered in the *Microsoft Office XP Resource Kit* at http://www.microsoft.com/office/ork/xp/default.htm.
- You have a test organization or test lab that verifies the results prior to deploying Office XP in your production environment.

Office XP Operating System Requirements

Table A.1 lists operating system support for Office XP.

Table A.1 Office XP Operating System Support

Operating System	SMS 2003 Support for Office XP
Windows XP Professional	Yes
Windows 2000 Server	Yes
Windows 2000 Professional	Yes
Windows Millennium Edition	No
Windows 98	Yes
Windows NT Server 4.0 SP6a or later	Yes
Windows NT Workstation 4.0 SP6a or later	Yes
Windows NT Server 4.0 SP6 or earlier	No
Windows NT Workstation 4.0 SP6 or earlier	No
Windows 3.1	No

(continued)

Table A.1 Office XP Operating System Support *(continued)*

Operating System	SMS 2003 Support for Office XP
Windows NT Server 3.51	No
Windows NT Workstation 3.51	No
Windows 95	No

If your clients are currently running any unsupported operating systems, you must upgrade the client to a supported operating system before installing Office XP. Alternatively, you can exclude from your Office XP deployment the computers that are running unsupported operating systems.

The approximate minimum hardware requirements for Office XP are:

- Pentium 133 or higher, PIII recommended
- 32 MB RAM, plus 8 MB additional per open application (128 MB is recommended), and 128 MB required for speech functionality
- 280 MB disk space for a restricted, minimum installation, but at least 600 MB is recommended

For more information about system and disk requirements, see the system requirements at http://microsoft.com/office/evaluation/sysreqs.htm.

Important Concepts and Issues

This section is an overview of the technology issues related to Office XP deployment with SMS, including:

- Package definition files.
- System Files Update.
- Multilingual User Interface Packs.
- Windows Installer versions.
- Windows Installer transform files.
- Windows Installer patches.
- How Office XP uses patches.
- Using the Windows Installer Install on Demand feature.
- Windows NT Workstation 4.0 low rights installation issues.
- Using the SMS administrative rights installation context.
- Office Resource Kit Tools.

Package Definition Files

A package definition file (.sms) is a template that contains the SMS components necessary to create and distribute software. SMS 2003 uses SMS packages to distribute software such as Office XP.

Office 2000 included a unique package definition file for every Office 2000 suite. Now, Office XP includes two main package definition files that you can use as a template to install any combination of Office XP suites. These files are updated as improvements are made. You can obtain the most recent package definition files from http://www.microsoft.com/Office/ORK/xp/JOURN/PDFWinXP.htm.

In Office XP, package definition file files have .sms extensions, and contain one .sms file for Office XP and one .sms file for Office XP multi-language deployments.

Table A.2 File Names and Descriptions

File name	Description
Off2002.sms	For all Office XP product suites
Lpk2002.sms	For the Office XP Multilingual User Interface Packs

The package definition file files are included in the Ork.cab file on the Office XP product CD. To access the package definition file files, install the Office XP Resource Kit, which is included in the ORK folder of the Office XP CD. You can also download the most recent package definition file files from the Office XP Resource Kit Toolbox on the Microsoft Office Web site at http://www.microsoft.com/office/ork/xp/default.htm.

System Files Update

Office XP requires a specific version level of a set of .dll files and other shared and system files. Before installing Office XP on computers running Windows NT Server 4.0 SP6a, Windows NT Workstation 4.0 SP6a, or Windows 98, Setup verifies that these key system files are current. If they are not, Setup updates the necessary files by running the System Files Update (SFU).

Key system files and components in the SFU include the following:

- Internet Explorer 5.01
- HTML Help
- Microsoft Data Access Components 2.5
- Microsoft Visual Basic® for Applications Runtime
- Microsoft Visual C Runtime
- Microsoft Foundation Class 4.2

By default, SFU installs Internet Explorer 5.01 on clients running Windows 98, Windows NT Server 4.0 SP6a, or Windows NT Workstation 4.0 SP6a. If you do not want to install Internet Explorer 5.01 on these clients, you can turn off the default Internet Explorer installation by using the Custom Installation Wizard. For information about the Custom Installation Wizard, see the "Customizing Your Office XP Installation" section later in this appendix, or see the *Microsoft Office XP Resource Kit* at http://www.microsoft.com/office/ork/xp/default.htm.

Note the following:

- Whenever SFU is run on a computer, a restart is required, even if no files were updated.
- Office XP does not install Internet Explorer on clients that are running Windows 2000 Professional, Windows 2000 Server, or later operating system versions, because Internet Explorer 5.01 is already included with these operating systems.
- Office 2000 Public Update 1 and later includes SFU. Upgrades from this version to Office XP do not have SFU issues.

Multilingual User Interface Packs

The Multilingual User Interface (MUI) pack provides plug-in language capability in Office XP and flexibility in customizing and installing the language files your organization needs. For example, you can customize Office XP to install with MUI pack files by chaining your installation, or you can customize and install language files after deploying Office XP.

For additional information about Office XP MUI Packs files, see the section on deploying Office XP internationally in the *Microsoft Office XP Resource Kit*.

Windows Installer Versions

Several different versions of Microsoft Installer are currently in use. Table A.3 is a list of the different versions and their delivery vehicles.

Table A.3 Windows Installer Versions

Windows Installer Version	Availability
Windows Installer 1.0	Included with Office 2000.
Windows Installer 1.1	Included with Windows 2000 Server and Windows 2000 Professional.
Windows Installer version 2.0	Included with Windows XP Professional. Also provided as a separate installable component from Microsoft.com for earlier versions of the operating systems.

Office XP requires Windows Installer 1.1 or later. If Windows Installer 1.0 is present on the computer, Office XP Setup automatically updates the program. If Windows Installer is not present on the computer, Office XP Setup calls Instmsi.exe (Windows 98) or Instmsiw.exe (Windows NT 4.0) to install it. Because Microsoft Windows 2000 includes Windows Installer 1.1, no Windows Installer update is required for computers running Microsoft Windows 2000.

Windows Installer version 2.0 fixes a number of problems in prior versions (including the ability to read source from an uncompressed source if installed from a compressed source, or vice versa). It is highly recommended that client computers be upgraded to this version prior to deployment of Office XP.

Windows Installer Transform Files

A Windows Installer transform (.mst) file provides configuration settings for a customized Office XP installation. A transform file contains information about components, features, setup properties, and changes that you can use to customize your Office XP installation. For example, if there are groups of computers in your organization that all require Microsoft Word and Microsoft Excel to be installed locally, but Microsoft Access and Microsoft PowerPoint® are set to install on demand, you can use a single transform file to configure the installation state you want for that entire collection.

The Office Resource Kit includes a tool called the Custom Installation Wizard that lets you change particular Office XP installation settings and create an .mst file. This file is then distributed with the source files for Office XP, so that the modified settings override the default values in Windows Installer when the installation is initiated on the client.

Windows Installer Patches

Windows Installer patch files allow the binary content of a Windows Installer installation to be changed without having to distribute an entirely new Windows Installer file. Office XP hotfixes and service packs use Windows Installer patch files.

Windows Installer uses a patch package to modify local or administrative installations. A patch package is a file with an .msp extension that can contain a modified version of any component in the original .msi file. A patch might contain a portion of the binary information for a particular component. Therefore, in most cases, the patch relies on the original installation .msi file to be available when the patch is applied.

There are two types of Windows Installer patches that Office XP can use:

- Administrative patches
- Client patches (also called standard patches)

Any particular fix encapsulated into a Windows Installer patch file has two different versions of the patch specific to each of these types. An administrative patch cannot be applied to a client installation, and vice versa. The following sections describe these patches in more detail.

Administrative Patches

Administrative patches apply the Windows Installer patch to the original Windows Installer source (the source on the SMS distribution points or, for most deployments, Windows Installer source servers). An advantage of this method is that it permits new client installations of Office to install the latest patch components without needing separate distributions of patches. A disadvantage of this method is that clients that have not been instructed to install the new patch cannot use this source for maintenance operations such as adding, removing, or repairing an installation until they have synchronized with the latest source. This is because the original Windows Installer source is changed (the package code of Windows Installer is changed). As a result, there is a risk that clients with Office installations can become orphaned, unable to run repair operations from the source, until they are instructed to apply the patch. However, correct deployment of patches using SMS can mitigate this risk considerably.

Client Patches

A client patch, also called a standard patch, is a distribution of the actual .msp file to individual client computers without changing the original source on the server. One advantage of this is that the server Windows Installer source remains valid for both patched and non-patched clients for maintenance operations. As a result, users do not experience problems using Office XP. One disadvantage of this method is that new client installations receive only the original non-patched Office version. They still need to receive the specific patch distributions to upgrade to the required Office component level. However, if the patches are being targeted using SMS software inventory information, then the upgrade occurs with no administrator intervention. However, additional client programs must run and a delay occurs in upgrading the client Office installation to its required state. As a result, client patching requires more SMS administrator overhead than administrative patching.

In most situations, the original Windows Installer source is required when you are applying a patch. For reliable patching, the original source should always be available when applying a client patch.

How Office XP Uses Patches

An Office update file is a packaged file that contains the .msp files for the update. Client patches also include the OHotFix patching program and .ini file. An Office update that is downloaded from the Office Web site is a file that contains one or more Windows Installer patch files and a file (ohotfix.exe) to apply them. The update file can be for an administrative update or a client update. The correct type of patch must be used for the appropriate installation. For more information about these issues, see
http://www.microsoft.com/Office/ORK/xp/JOURN/Cliupdt.htm.

Windows Installer patches are used for distributing both Office XP Public Updates and service pack components. In fact, an Office XP service pack includes all previously released Public Update patches, in addition to new fixes. However, service packs are particularly important for patch management because they apply a new baseline for the installed components against which future Public Updates (individual patches) are applied. To apply an individual Office public update, the appropriate service pack for the patch must already be applied to the original Office source.

Using the Windows Installer Install on Demand Feature

Windows Installer Install on Demand allows Office XP features to be installed when the user selects them. Install on Demand is configured in the Custom Installation Wizard by selecting the **Install on first use** option for the specific applications.

The following are advantages of using Install on Demand:

- Less disk space is required initially on the client computer.
- Peak network bandwidth utilization is potentially decreased because Windows Installer components are copied over a longer period of time instead of simultaneously.

A disadvantage of using Install on Demand is that the original Windows Installer source must be accessible at the time that the Office XP feature is selected. Install on Demand is not a good choice for laptop computers when the Windows Installer source is not stored locally on the computer.

Important

Install on Demand applications generate status messages when SMS runs the Office XP installation for the first time. However, SMS does not generate status messages when the user initiates the Install on Demand installation (after the initial installation) because later Office XP component installations are not initiated via the SMS client.

To create desktop icons, the Install on Demand method requires Active Desktop to be installed on the client; however, it is not mandatory that Active Desktop be turned on. If Active Desktop is not installed, the Windows Installer shortcuts might not be installed, and install-on-demand functionality is not available. Also, all clients must be running Microsoft Internet Explorer 4.01 or later. Clients running Windows 98, Windows 2000 Server, Windows 2000 Professional, or Windows XP have this functionality built in.

You can customize and extend the basic methods of installing Office XP on clients to help support your specific deployment needs. After you run Install on Demand Setup, you can send additional Office XP Setup command lines to clients to trigger the installation of a specific Office XP program. This can be helpful if you want to stage your Office XP deployment.

For example, you might want to install the full version of Word and Excel on clients, but make PowerPoint and Access available for installation on first use. By setting some of the programs to install on first use, the network bandwidth required at initial deployment time is reduced, and setup time for users is faster. Later, if you want complete versions of PowerPoint and Access installed on all clients, you can send an additional setup command line program to accomplish this using SMS.

For additional information about Office XP features and Install on Demand, see the *Microsoft Office XP Resource Kit* at http://www.microsoft.com/office/ork/xp/default.htm.

Windows NT 4.0 Low Rights Installation Issues

Low rights installation refers to the situation where the logged-on user does not have sufficient rights to complete an installation of Office XP. This is particularly problematic for SMS when the installation requires a restart (in the case of SFU installations) where the security context of the installation is lost during the restart. This scenario occurs for Windows NT 4.0 computers installing Office XP if both of the following are true:

- The logged-on user is not an administrator on the computer
- The Office XP installation is a clean install or an upgrade from a version of Office prior to Office 2000 Public Update 1 (because Office 2000 Public Update 1 includes SFU).

You can use the Elevated Rights Deployment Wrapper (Run_once.exe) to install Office XP in this scenario. The Elevated Rights Deployment Wrapper is typically used to deploy Office XP to a locked-down computer that is running Windows NT Workstation 4.0. You can use the Elevated Rights Deployment Wrapper with applications that require a restart during installation and elevated user rights following the restart. With this tool, the post-restart work can be integrated with SMS for both elevated rights and status reporting. Check the Systems Management Server Downloads Web site at http://www.microsoft.com/smserver/downloads/20/default.asp for this tool.

Using the SMS Administrative Rights Installation Context

The **Run with administrative rights** option on the **Environment** tab of the **SMS Program Properties** dialog box grants Administrative credentials to the logged-on user even if the user is a low rights user. This option ensures that low rights users can complete Office XP installations. The Administrative context is maintained until the computer restarts. Special procedures are required for installing the SFU components in a low-rights scenario because a system restart is required.

Office Resource Kit Tools

This section describes the following tools for deploying Office XP with SMS that are included in the *Microsoft Office XP Resource Kit*:

- Custom Installation Wizard
- Office Profile Wizard

Custom Installation Wizard

You can use the Custom Installation Wizard to make changes to the master installation in an .mst file without altering or resending the original Office XP source files. Because the original package is never altered, you can create a different transform file for every installation scenario you have. When you run setup with both the package and the transform file, Windows Installer applies the transform file to the original package, and Setup uses your .mst file to obtain the configuration for the installation. All of this can be done without building a new package, which minimizes bandwidth usage.

The Custom Installation Wizard is automatically installed on your computer when you install the *Office XP Resource Kit*. You can use the Custom Installation Wizard to visually build your transform files.

Office Profile Wizard

You can use the Microsoft Office Profile Wizard to save configuration settings for Office XP applications. These settings are saved in a profile settings file (.ops), which is a snapshot of registry settings and related files for an Office user configuration. Using the Profile Wizard, deployment experts can further configure the deployment of Office XP to include a wide range of settings. These settings are more detailed than the Custom Installation Wizard. For example, the Profile Wizard allows an administrator to configure a test system. The Profile Wizard then collects the registry information as the administrator configured it, and the resulting .ops file is embedded in an .mst file.

Using the Profile Wizard, the administrator can create a virtual image of the Office XP installation as it should be configured when it is deployed. With the Profile Wizard, the administrator captures all settings that represent a combination of registry settings and files providing a specific user with a customized working environment. Typical customizations are the use and alignment of toolbars, common menu options, and language or tool visibility. Many customizations are set in the **Tools, Options** or **Tools, Customize** menu option of most Office applications.

Office XP CD and Administrative Installation Source Issues

The type of Windows Installer source used for an Office XP installation depends on how you install Office XP. When Office XP is installed using Setup from the CD or a network share, it uses the Windows Installer compressed source. When Office XP is installed from an administrative installation point created by running **setup /a**, it uses an uncompressed source.

The Office Windows Installer source types from these two types of sources are not compatible with each other. As a result, if you use a different source type after installation, your installation either fails or Office XP is uninstalled and then reinstalled from the new source type. This problem typically occurs when Office XP has been installed directly from an Office XP CD, but then an update to the Office XP installation is distributed that references an administrative installation point source. This issue is fixed with Windows Installer version 2.0.

You can avoid this incompatible source problem by enforcing the use of one type of Office XP source in the enterprise. For example, if an administrative installation of Office XP is being used within an environment managed by SMS, then give users CDs of the image that they can install manually. For more information, see the Office deployment section in the Microsoft Office Resource Kit. Users should then be discouraged from using retail versions of Office XP installations. In addition, computers with the retail version of Office XP (for example, vendor or non-corporate owned computers) should not be deployed in the SMS hierarchy.

It is technically possible to maintain both a standard non-compressed and an administrative compressed version of Office XP on the SMS distribution points, and then direct clients to the appropriate source based on the installed Office source type. However, this is difficult to enforce reliably and is beyond the scope of this appendix. A better approach is to periodically distribute a command for the client to synchronize the installed Office XP source with that on the installation point. This procedure is the same as that described for synchronizing following the patching of the installation point source with a Windows Installer patch:

1. Create a program for each site called **Office Synchronization Program**.
2. Add the command line:

   ```
   setup.exe /fvm
   ```

3. For setup command-line options, see http://www.microsoft.com/office/ork/xp/appndx/setopt.htm.
4. Create an advertisement for each site to advertise this program.

Deploying Office XP in an Organization

This section describes how to use SMS 2003 to deploy Office XP in a large, diverse corporate organization containing multiple primary and secondary SMS sites.

This scenario assumes that you, as the administrator, are familiar with the Custom Installation Wizard and issues regarding Office XP deployments and supported installation methods. For technical background on relevant issues, see the "Important Concepts and Issues" section earlier in this appendix. You should also be familiar with SMS software distribution, collections, queries, and reporting.

Business Requirements

The scenarios and instructions outlined in this appendix for using SMS 2003 to deploy Office XP assume that the following business requirements, site, and client configurations are present. These requirements are a superset of the most common requirements for any particular enterprise.

The SMS administrator in the sample scenario must be able to:

- Perform a full installation of Office XP on local computers.
- Perform a partial installation of Office XP on local computers, with Install on First Use available for other Office XP programs.
- Use customized and additional templates.
- Use specific preconfigured options in some of the Office XP programs.
- Install Office XP on all new computers that connect to the network.
- Install Office XP at multiple sites.
- Upgrade Office XP installations without user intervention.
- Minimize the overall administrative requirements for Office XP deployment.
- Report the distribution results, and present an ongoing management report of computers that do and do not have Office XP installed.
- Maintain the Windows Installer support for repairing Office XP core files to reduce support costs.
- Reduce drive space usage by reducing the number of Office XP distribution points after Office XP deployment.

Enterprise Configuration

SMS 2003 is configured in this sample scenario as follows:

- One central site has ten primary sites.
- Each of the primary sites has 40 secondary sites attached.
- Each secondary site is separated from its respective parent site by a WAN link.

- Client and site traffic is minimized architecturally. Each SMS site is contained within one continuous fast link. Slow links are managed by separating sites, and none of the sites span a slow link. Throttling and sending times are controlled to prevent SMS from using significant bandwidth during prime operating hours.
- SMS 2003 is deployed in the enterprise.

Corporate migration to the Active Directory structure for the corporation is completed.

Client Configuration

The company in this scenario has a mix of client operating systems configured as follows:

- Clients running Windows 98, Windows NT Workstation 4.0 SP6a, and Windows 2000 Professional.
- Users at some of the sites do not have local administrator rights on their clients that are running Windows NT Workstation 4.0 SP6a and Windows 2000 Professional.
- The majority of users are upgrading from Office 2000 Public Update 1, but some users are upgrading from Office 97.

Planning the Deployment

This section assumes that you have successfully tested your deployment in a lab using hardware and software identical to your enterprise. After you have tested your deployment in a lab environment, you are ready to begin the Office XP deployment in the first production-pilot environment. This first deployment should be a small pilot group, which you can carefully monitor, and adjust plans according to findings.

Before deploying Office XP in a production environment, you must:

- Consider basic planning issues.
- Determine the systems and sites that will be upgraded.
- Determine SFU requirements.
- Plan for clients without administrative credentials.
- Determine which clients require software upgrades prior to installing Office XP (through software inventory and queries).
- Plan installation options.
- Plan the strategy for collections and program advertisements.
- Prepare and customize the Office XP installation.

Basic Planning Considerations

- Verify that the clients have the hardware and software required to run Office XP. See "Office XP Operating System Requirements" earlier in this appendix.
- Know the terminology and processes of SMS 2003 software distribution, the Office XP Custom Installation Wizard, and the Office Profile Wizard.
- Review the features and options available with Office XP. Ensure that you properly implemented the configuration (.mst file) you built using the Office XP Custom Installation Wizard and Office Profile Wizard.
- Determine which Office XP features you want to install, and which systems you want to receive each group of features. Each group will receive its own transform file.
- Decide whether to do an attended or unattended installation, so that you can properly configure the program and advertisement properties.
- Decide if your installation is going to be mandatory or optional. If the installation is mandatory, you can allow users to install it before the scheduled time. If users do not install the program early, the program automatically runs at the scheduled time.

  Unattended mandatory deployments offer potentially increased benefits to the enterprise, because users are not present or needed during installation. Mandatory installation also ensures standardization within the business unit.

Determine the Systems and Sites That Will Be Upgraded

In this example scenario, you have decided to deploy Office XP to the central site (CTL) and the remote site (RMT) across a WAN link, using two different custom configurations of Office XP. CTL is the parent site of RMT. Productivity needs at both sites have determined the requirement for the upgrade to Office XP.

Determine SFU Requirements

Office XP Setup installs the SFU by default. If you do not want the SFU installed, select a program for your Office XP package that has **/nosp** in the command line. For example, the **Custom (quiet)** program specifies **/nosp** in the command line and does not install SFU. After installing the SFU, the client is restarted to complete the SFU installation.

A typical scenario for using SFU might be as follows. You have two computers. One runs Windows 98, and the other runs Windows 2000 Professional. Both have Office 2000 installed. The computer running Windows 98 must receive the SFU to install Office XP, but the computer running Windows 2000 Professional does not require the SFU. If you run a command line on both computers that does not exclude the SFU routine, the Windows 98 computer receives the SFU, but the SFU is not reinstalled on the computer running Windows 2000 Professional. However, at the end of the first phase of the SFU, a restart is included in the command line, and this affects both computers.

Clients That Require the SFU

The program must include the setup command line to install the SFU on the following clients:

- Clients running Windows 98 on which neither Office 2000 Public Update 1 nor Office XP was installed.
- Clients running Windows NT Workstation 4.0 SP6a, or later, on which neither Office 2000 Public Update 1 nor Office XP was installed.

Important

If you have clients running Windows NT Workstation 4.0 SP6a with users who do not have administrative credentials on the local computer, use the Elevated Rights Deployment Wrapper (Run_once.exe) to install Office XP. See the Systems Management Server Downloads Web site at http://www.microsoft.com/smserver/downloads/20/default.asp for this tool.

Clients That Do Not Require the SFU

Setup does not install the SFU on clients running the following operating systems, because these operating systems already have the required versions of key system files:

- Windows 2000 Server, Windows 2000 Professional, or Windows XP Professional.
- Any clients running Windows NT Server 4.0 SP6a, or later, or Windows NT Workstation 4.0 SP6a, or later, on which Office 2000 Public Update 1 or Office XP has been previously installed.
- Any clients running Windows 98 on which Office 2000 Public Update 1 or Office XP has been previously installed.

Plan for Clients Without Administrative Credentials

The SFU and restart phases of the Office XP installation require that the logged-on user have administrative credentials on the local computer before and after the computer restarts. This might or might not be an issue, depending on your client install base. This section explains the requirements for each operating system supported by Office XP.

Clients Running Windows 2000 and Windows XP

These clients do not need the SFU. Office XP installation requires administrative credentials, therefore, you must specify administrative credentials in the SMS program for low-rights users. This setting is included in the package definition files.

Clients Running Windows 98

If Office XP Setup determines that a client requires the SFU, the SFU is installed, a restart is initiated, and then Office XP is installed.

Clients Running Windows NT 4.0

Office XP installs without special consideration on clients running Windows NT Workstation 4.0 or Windows NT Server 4.0 if either of the following is true:

- The client does not need the SFU.
- The client needs the SFU, but the logged-on user has administrative credentials on the local computer to allow SFU to complete.

If the computer running Windows NT 4.0 requires the SFU, and the account used for installation does not have administrative credentials on the local computer, then the Office XP installation fails.

Important

You can use the Elevated Rights Deployment Wrapper (Run_once.exe), described earlier in this appendix, to deploy Office XP to low-rights Windows NT 4.0 clients that require the SFU.

For computers running Windows NT 4.0 with low-rights users, it is recommended that you upgrade the client to Windows 2000 or later before installing Office XP.

Office 2000 includes a script that was intended to help install Office 2000 on computers where users did not have local administrative credentials. This script has had several problems with supportability. The script is not supported in Office XP.

Determine Which Clients Require Upgrades Prior to Installing Office XP

In the sample scenario, the administrator and the corporate decision-maker decided to upgrade the operating systems that are not supported by Office XP as show in Table A.4.

Table A.4 Actions Based on Client Environment

Client environment	Action
Windows NT 4.0 SP6a users without local administrative rights (Office 2000 Public Update 1 is not already installed)	Upgrade clients that do not already have Office 2000 Public Update 1 installed to Windows 2000 Professional or Windows XP Professional.
Windows NT 4.0 SP6a without local administrative rights (Office 2000 Public Update 1 is already installed)	Clients with low rights running Windows NT 4.0 SP6a that are upgrading from Office 2000 Public Update 1 do not need the SFU, and Office XP can be directly deployed.
Windows XP and Windows 2000 Professional	No action needed. Clients running Windows XP Professional, and Windows 2000 Professional already have the proper files and do not require the SFU.
Windows 98	No action needed. Clients running Windows 98 allow the SFU installation to complete successfully after the restart.

Plan Installation Options

You have the following options when installing Office XP:

- Use the SMS administrative context with the programs included in the package definition file.
- Use the SMS administrative context with Install on Demand.

Plan the Strategy for Collections and Program Advertisements

The Office XP package definition file creates the Microsoft Office XP Applications 10.0 English package, a package that contains eight previously created programs. These programs are templates to deliver Office XP to different clients or under different situations. With SMS 2003, you can create collections based on whether or not clients need the SFU, and advertise the appropriate programs to each collection. For example, you might want to create collections as shown in Table A.5.

Table A.5 Collections Based on SFU Requirement

Collection	Client operating system	Comments
No SFU Collection	Windows 2000 and Windows XP	These computers do not require the SFU and, therefore, do not require a restart.
SFU Collection	Windows 98 and Windows NT 4.0 SP6a (with administrator rights on which Office 2000 Public Update 1 is not already installed	These computers require the SFU and, therefore, require a restart.

Then you can advertise the appropriate programs to each collection, as shown in Table A.6.

Table A.6 Target Collections

Installation program	Target collection
Custom (unattended)	Advertise this program to the No SFU Collection.
Custom including SFU (unattended)	Advertise this program to the SFU Collection. The command line includes a restart to the systems after running the SFU program.

Note

The Office Setup command line with the SFU runs successfully on computers that do not require the SFU. SFU installation does not occur, but those computers still restart. To make the deployment simpler, you can target all computers by using the SFU and restart instructions in the program. Installation is successful on all clients that can install Office XP. However, clients that did not require the SFU are restarted, even though a restart was not required.

The following table describes how each SMS program included in the package definition file installs the SFU on the supported operating systems, where:

Targeted indicates that the client receives the program advertisement.

Not targeted indicates that the client does not receive the program advertisement.

Fails indicates that the Office XP installation fails.

Table A.7 Office XP Programs to Use with Each Supported Operating System

Installation Program	Operating System				
	Windows 98	Windows NT 4.0[1]	Windows NT 4.0[2]	Windows 2000	Windows XP
Custom (quiet)	Not targeted	Not targeted	Not targeted	Targeted	Targeted
Custom including System Files Update (quiet)	Targeted	Fails	Targeted	Not targeted	Not targeted
Manual	Not targeted	Not targeted	Not targeted	Targeted	Targeted
Manual including System Files Update	Targeted	Fails	Targeted	Not targeted	Not targeted
System Files Update only (quiet)	Targeted	Fails	Targeted	Not targeted	Not targeted
Typical (quiet)	Not targeted	Not targeted	Not targeted	Targeted	Targeted
Typical including System Files Update (quiet)	Targeted	Fails	Targeted	Not targeted	Not targeted
Uninstall	Targeted	Targeted	Targeted	Targeted	Targeted

1. User logging on after the SFU restart does not have local Administrator rights on the client.
2. User logging on after the SFU restart has local Administrator rights on the client.

Prepare and Customize the Office Source

The next step in the deployment process is to prepare the office source files for use by SMS and create any customization files that are required. If you are deploying a typical Office installation, then it is not necessary to create your own customization files (.mst and .opw files). These customization files are referenced by the SMS program objects.

Deploying Office XP

The following is a summary of the steps for deploying Office XP, each of which are fully discussed in subsequent sections. Some steps, such as Office XP source customization, might not be necessary in your deployment, but most deployments follow this basic pattern.

1. Create the administrative installation point.
2. Create the transform file using the Office Custom Installation Wizard.
3. Create an .ops customization file using the Office Profile Wizard.
4. Create SMS software distribution objects.
   - Import the package definition file and create the Office XP package.
   - Create programs for Office XP Install on Demand.
   - Create the collections.
   - Assign distribution points to the Office XP package.
5. Check the progress of your installation.
   - Monitor the installation.
   - Conduct inventory and reporting of the Office XP installations.
   - Run a query to confirm that Office XP was successfully deployed.

Step 1: Create the Administrative Installation Point

An administrative installation point is a server share that contains all the expanded files you need to install Office XP by using SMS 2003. When SMS 2003 packages are distributed, the files in the administrative installation point are copied to distribution points. Restrict access to these source files through share permissions or NTFS file system privileges.

Using the administrative installation point can eliminate interactions at the client, such as prompting for a product key. Valid licensing is required, and all licensing requirements apply when you deploy Office XP.

The following steps describe how to create an administrative installation point. For more information, see the *Microsoft Office XP Resource Kit*.

1. Create a share on a network server for the administrative installation point. This share will be the source file location for the Office XP package. To avoid a package source update spanning a WAN, you should create the share on each site server that will send the package. The transform files created by the Custom Installation Wizard should also be placed in the root of this share, along with the .ops file created in the Office Profile Wizard used to further customize Office XP programs.

   Important

   The network share must have at least 700 MB of available disk space. Increased space requirements might arise during the process, depending on configuration.

2. Connect to the server share using an account that has write access to the share. The computer must be running a supported operating system, such as Windows XP Professional, Windows 2000 Professional, Windows NT Workstation 4.0 SP 6a, or Windows 98.
3. On the **Start** menu, click **Run**, and then run **setup /a** from the root of the Office XP product CD.
4. Enter the organization name that you want to define for all users who install Office XP.
5. Enter the server and folder you created as the installation location. The SMS administrator selected a local path, to avoid unnecessary network traffic. If a remote path is selected, then the administrative installation is copied on the network. In our scenario, the path is C:\OfficeXP.
6. Enter the 25-character product key and click **Next**. You must enter a valid, purchased product key when you create the administrative installation point. Users who install Office XP from this administrative image do not have to enter the product key when they install Office XP or start an Office XP program for the first time.
7. Accept the end-user license agreement and click **Install**. By accepting the agreement, you are accepting on behalf of all users who install Office XP from this administrative installation or any SMS distribution point the SMS administrator creates from this administrative installation.

Setup expands and copies the files from the Office XP CD to the administrative installation point, extracts the compressed cabinet (.cab) files, and creates a hierarchy of folders in the root folder of the share. The SFU files are automatically included during an administrative installation.

By creating multiple transform files, you can also install different Office XP configurations to different groups of users from the same Office XP source file location (also called an administrative installation point).

You can use a transform file to create any kind of customization in an Office XP command-line by using the Custom Installation Wizard. You can create transform files for:

- Adding resilient sources.
- Creating customized programs or configurations for different sets of users.
- Selecting the Office XP features that you want to install.
- Modifying shortcuts.
- Customizing options for Microsoft Outlook®.
- Adding custom files to the installation.
- Customizing the SFU settings, including Internet Explorer 5 settings, as needed.
- Customizing Office XP Multilingual User Interface Packs.

Step 2: Create the Transform File by Using the Office CIW

Transform files contain instructions for customizing Office XP. To customize the Office XP installation, you can create a custom transform file by using the Microsoft Office Custom Installation Wizard (CIW).

Table A.8 describes some feature states that you can configure using the CIW, and that are displayed on the **Set Feature Installation States** page of the CIW.

Table A.8 Feature States

Feature State	Description
Install on demand	This feature state is indicated by the number "1" next to the component name, and means that the component is installed on first use.
Install to hard disk	This feature state is indicated by a disc drive icon next to the component feature, and means that the component is installed on the client hard disk during the initial installation.
Not Available	This feature state is indicated by a red "X" next to the component name, and means that the component is not installed.

In the example scenario, you decide that there will be one transform file per site, because all the distribution points within a site will be local. As a result, any distribution points will be local, and traffic associated with repairs and install-on-demand applications will not cross WAN links in the future. You also create a hidden custom share for the Office XP source called OfficeXP$.

To distribute the package properly, you must record key information for each site. Table A.9 contains information about the CTL site.

Table A.9 CTL Site Distribution Points

Distribution point name	Is this a permanent distribution point?	Network path that SMS uses to access and install Proplus.msi, .mst files, .ops files, and installation files
Site1dpa	Yes	\\site1dpa\OfficeXP$
Site1dpb	Yes	\\site1dpb\OfficeXP$
Site1dpc	No	\\site1dpc\OfficeXP$

You have now determined the location, administrative installation point, and share for the installation files of Office XP. Next, you must build the files to configure the installation.

You create the transform file by using the Custom Installation Wizard. Repeat the following steps for each transform file.

1. Run the Custom Installation Wizard by clicking **Start**, pointing to **Programs**, pointing to **Microsoft Office Tools**, pointing to **Microsoft Office XP Resource Kit Tools**, and then clicking **Custom Installation Wizard**. The administrator runs the Custom Installation Wizard on a computer that has access to both the administrative share (where the .mst files are copied) and the Proplus.msi file that installs Office XP.
2. Specify the path of the .msi file to open (for example, D:proplus.msi), and click **Next**.
3. Create a new transform file by selecting **Create a new MST file**, and then click **Next**.
4. Specify the path of the new transform file (for example, C:Documents and Settings\user1\my documents\ new custom setup file for CTL Site.mst), and then click **Next**. Long file names are allowed, so give the transform file a descriptive name (including the site name and configurations). This provides an easy way to reference the file later.
5. Specify the installation location (<*ProgramFiles*>\Microsoft Office) and the company name, and then click **Next**.
6. Set the feature installation states and verify that the Office XP installation meets the needs of the users. The **Installed on First Use** option on the Set Feature Installation States page helps minimize setup time for the user, and balances the traffic associated with installation. For example, if you set the feature state for Microsoft Access for Windows to **Installed on First Use**, the traffic associated with Access is deferred until a user on the system runs Access, or until the user runs an Access file. When applications that are set to **Installed on First Use** are first activated, there is a delay while the installation runs.
7. Using the table that was completed when creating the transform file for the clients, enter the paths of the installation points. The Custom Installation Wizard does not validate the paths. These paths are entered so that the resilient sources are displayed at the top of the list. The administrator does this so that when the resilient sources are removed from \\site1dpc, clients seeking to install or repair from this transform file will go to the existing site first, instead of failing against \\site1dpc, and then succeed in reaching the resilient sources (\\site1pda and \\site1dpb).

8. Accept the changes and create the .mst file for the Office XP installation that installs Office XP with Word and Excel locally and installs the other applications and tools on first use.
9. Include any additional configurations that you made with the Profile Wizard. These are embedded in the .mst file. You can use the Profile Wizard to configure the options contained within the **Tools** menu, in addition to other options to customize Office XP to your requirements. For more information, see the *Microsoft Office XP Resource Kit*.
10. Review the summary, and exit the CIW.
11. Copy the transform file into the root of the administrative installation source (which was created by running **setup /a**).

Important

All transform files should be included in the root of the administrative share prior to assigning the SMS package to any distribution points. This prevents repeated replication of files to all designated sites and distribution points.

Step 3: Create an .ops Customization File by Using the Office OPW

You can further customize an Office installation using the Office Profile Wizard in the *Microsoft Office Resource Kit*. The Office Profile Wizard takes a snapshot of the registry settings for an existing Office XP installation and saving the settings to an .ops file. This file is then saved in the root of the Office source files in the same location as the .mst file.

Step 4: Create SMS Software Distribution Objects

If you need customized SMS programs that are different from the ones included in the Office XP package definition file, you can modify the programs created by the .sms file after you create the package.

With SMS 2003, you can also create SMS packages by importing the appropriate .msi file (such as proplus.msi). However, this creates different programs.

Note

The Custom programs within SMS might reference a transform file to allow those installations to be highly configured by the transform file. There is also an SMS program created by the .sms file named Typical. The Typical program does not use transform files.

To create the software distribution objects, do the following:

1. Import the package definition file (.sms file), and create the Office XP package.
2. Create programs for Office XP Install on Demand.
3. Create the collections.
4. Assign distribution points to the Office XP package.

Step A: Import the package definition file, and create the Office XP package.

To create a package from a package definition file, you can use the Create Package from Definition Wizard or the Distribute Software Wizard. Both wizards create a package by importing a package definition file, but the Distribute Software Wizard includes the following additional functionality:

- Enabling and disabling distribution points for the package
- Creating an advertisement for the package

► To create a package using the Package Definition Wizard:

1. Expand the Ork.cab file that is included in the ORK folder of the Office XP CD to access the package definition file. Click the **Ork.cab** file, click **Extract**, and then choose a destination folder.
2. Start the SMS Administrator console and double-click **Site Database**.
3. Right-click **Packages**, point to **New,** and then click **Package from Definition**.
4. On the Create Package from Definition Wizard page, click Next.
5. The Package Definition page is displayed; however, the Off2002.sms file does not appear in the list because it has not been imported into the SMS Administrator console. Click **Browse**.
6. In the folder that contains the .sms file you extracted, click **Off2002.sms,** and then click **Open.**
7. Click **Office XP Applications**, and then click **Next**.
8. On the Source Files page, select the **Always obtain files from a source directory** check box, and then click **Next**.
9. On the Source Directory page, specify the location of your administrative installation point. You must specify a local path on the site server or a network path to the administrative installation.
10. Click **Browse** to locate the source location for the Office XP administrative installation (**C:**\OfficeXP in the example), and then click **Next**.

Verify that the information is correct, and then click **Finish** to create the package named Microsoft Office XP applications 10.0 English.

Note

Updated package definition files for Office XP are available in the *Microsoft Office XP Resource Kit* at http://www.microsoft.com/Office/ORK/xp/JOURN/PDFWinXP.htm.

Step B: Configure programs for Office XP Install on Demand

By importing the Off2002.sms file and creating the package from it, you now have eight default programs created under the package named **Microsoft Office XP applications 10.0 English**. You can use these different programs in the collections that target the separate installation requirements for systems receiving Office XP.

There are two programs the SMS administrator uses in this phase of the deployment. Both programs are unattended, so that the installation can be fully managed by the SMS administrator, and users are not prompted for installation input.

Custom (Quiet)

This program is used on all clients that do not require the SFU and the associated restart, including computers running:

- Windows XP Professional.
- Windows 2000 Professional.
- Windows 2000 Server.
- Windows 2000 Advanced Server.
- Windows NT Workstation 4.0 SP6a computers with Office 2000 Public Update 1.
- Windows 98 with Office 2000 Public Update 1. Computers that use this program to install Office XP do not require a restart.

Custom including System Files Update (Quiet)

This program is used to install Office XP on all clients that require the SFU. Completing the SFU installation requires a restart. The computers that use this program in their advertisement can include computers running:

- Windows 98 without Office 2000 Public Update 1 installed.
- Windows NT Workstation 4.0 SP6a without Office 2000 Public Update 1 installed. (Users must have local Administrator permissions to complete the SFU before Office XP installs.)

The administrator must decide Install on Demand options for clients in each site, as shown in Table A.10.

Table A.10 Installation States

Installation states	Issues to consider
For clients at the CTL (central) site	You can create a separate transform file for each different installation state of Office XP that is required. You might edit existing transform files and save them under separate, descriptive names. These new transform files should be placed in the root of the source directory (C:\OfficeXP in the example), along with any accompanying Office profile files (.ops) created in the Office Profile Wizard. Complete this prior to creating the package's distribution points, because it minimizes traffic associated with the package source files.

(continued)

Table A.10 Installation States *(continued)*

Installation states	Issues to consider
For clients at the RMT (remote) site	Use the Custom Installation Wizard to efficiently edit the existing transform (.mst) files to match the installation states of the CTL site's requirements. The resilient sources, which appear in the Custom Installation Wizard on the Additional Servers page, should be changed to point to the distribution points that are local for the clients at the RMT site. This ensures that the RMT clients both repair and install on demand from the share of the local distribution point within the RMT site.

The administrator must perform the following tasks for each transform file:

1. In the Custom Installation Wizard, on the General page, enter the transform (.mst) file name in the command line.
2. Specify a descriptive name for the file, such as *Word and Excel Local Access and Powerpoint IOD.mst*.
3. Specify the transform file name in both Custom (quiet) and Custom including System Files Update (quiet) programs.

Step C: Create the collections

Creating collections is the first step toward deployment. A collection is created in SMS by running a query, which includes systems based on their compliance with the query rules. For example, your query should contain a minimum RAM requirement that you have tested and verified to be successful for your users. If you make the RAM requirement part of the query rules for the collection, only systems with sufficient RAM are targeted for the Office XP installation. The Office XP package, with the related program and correct transform file, are distributed based on the following criteria:

- The Office XP configuration (.mst file) the computer requires. Those computers with common configurations receive the same .mst file in the command line of the SMS program.
- The requirements of the computers before Office XP is installed, for example, whether those computers require the SFU and restart to be a part of the installation routine.

Determine which computers in your sites should receive Office XP. These computers are grouped into SMS collections. You can use several methods, depending on the business rules your organization has in place. In the example, it is assumed that all clients currently running Office 2000 Public Update 1 are upgraded, and computers running Windows NT 4.0 are excluded. Computers with less than 64 MB of RAM and less than 500 MB of free disk space are also excluded.

Create dynamic collections that encompass all systems that you want to receive Office XP. All computers in this collection require Office XP with Word and Excel installed locally, while Access and PowerPoint are installed on first use. When new systems come online or meet these criteria, the Office XP package is also assigned to them. This way, the SMS administrator adheres to the company's corporate policy to have new systems install Office XP.

The query can be written to create collections based on the criteria in Table A.11.

Table A.11 Criteria for Collections Queries

Collection	Site Name	Operating system	Current Office Version	At least 64 MB RAM	At least 500 MB disk space
C1	CTL	Windows XP Professional	Any	Yes	Yes
C1	CTL	Windows 2000 Professional	Any	Yes	Yes
C1	CTL	Windows 98	Office 2000 Public Update 1	Yes	Yes
C2	CTL	Windows 98	Any except Office 2000 Public Update 1	Yes	Yes

When creating the query, it is important to use your knowledge of the separate setup routines described earlier in this appendix. Computers running Windows 98 without Office 2000 Public Update 1 require the SFU and have a different command line than your other clients. Therefore, separate the systems needing different programs.

These dynamic collections are targeted for different advertisements that point to two separate program command lines. The transform file you use is the same, because both the systems in the collections C1 and C2 require the same Office XP configurations. However, because the systems in collection C2 require the SFU and associated restart, the SMS administrator in the example separated the collections. That way, all clients running Windows XP Professional, Windows 2000, and Windows 98 with Office 2000 Public Update 1 do not require a restart.

Step D: Assign distribution points to the Office XP package

Planning for the distribution points is essential to providing local distribution points for the Office XP package. Also, if you intend to use resilient sources, you should know both the distribution points and the custom share name that SMS uses for the package. This is the path that you should have entered on the **Additional Servers** page in the Custom Installation Wizard.

You can assign distribution points and create an advertisement by using one of the following:

- The SMS Administrator console
- The Distribute Software Wizard. When you are ready to deploy Office XP to your clients, you can run the Distribute Software Wizard to create the distribution points and the advertisement targeted at all systems in your organization.

After Office XP deployment, you might plan to reduce the number of distribution points. At least one distribution point remains locally at each site so that Office XP Windows Installer can automatically repair a corrupt core file by accessing a local distribution point. Also, application installations occurring on first use continue to have access to these distribution points.

> **Note**
>
> By design, SMS clients receive only information regarding distribution points in their own site.

Each of these location-specific transform files is then included in a new program created within the Microsoft Office XP applications 10.0 English package. The command line entry on the program's **General** tab includes the direction to the newly created, location-specific transform files.

All of this planning should be done in advance, because these new .mst files must be placed in the root of the source location (C:\OfficeXP in the example). You can change the list of additional servers using the Custom Maintenance Wizard and use SMS to distribute the update after installing Office XP. If you deploy this package to distribution points, and then add the new .mst files to the source, the entire source directory must be sent to the distribution points during the next package refresh of the distribution points. Sending the entire source directory produces considerable network traffic, because the Office XP installation files, which are approximately 625 MB, would be redeployed to each distribution point.

Assign the distribution points for the Microsoft Office XP applications 10.0 English package. For example, all of the previously determined distribution points could be used:

- Site1dpa
- Site1dpb
- Site1dpc

> **Note**
>
> These distribution points should be within the site, which is defined within a continuous fast link. If a distribution point is separated from the systems that install or repair Office XP, then a second location-specific transform file should be created that lists only the distribution points connected by a fast link to the clients. Architecturally, SMS is designed to support this model, where each distribution point for a site resides on the same fast link as the clients it serves.

► To create an advertisement and assign SMS distribution points using the Distribute Software Wizard

1. In the SMS Administrator console, navigate to **Packages**, right-click your package**,** point to **All Tasks**, and then click **Distribute Software**.
2. On the Distribute Software Wizard page, click **Next**.

3. The Package page is displayed. Select the **Distribute an existing package** check box, click **Office XP Applications**, and then click **Next**.
4. On the Distribution Points page, select the distribution points you want for the Office XP distribution, and then click **Next**.
5. On the Advertise a Program page, click **Yes,** and then click **Next**.
6. On the Select a Program to Advertise page, click the program that you want to include in this advertisement, and then click **Next**.
7. On the Advertisement Target page, select the **Advertise the program to an existing collection** option, click **Browse** to select a collection, click **OK**, and then click **Next**.
8. On the Advertisement Name page, click **Next**.
9. On the Advertise to Subcollections page, select the appropriate option, and then click **Next**.
10. On the Advertisement Schedule page, select the appropriate schedule, and determine whether you want the advertisement to expire. If you are using the distribution points as resilient sources, select the **No. This advertisement never expires** check box, and then click **Next**.
11. On the Assign Program page, select the appropriate option, and then click **Next**.
12. On the Completing the Distribute Software Wizard page, verify the information under **Details,** and then click **Finish** to start the distribution.

Following the standard method for advertising to a collection, the SMS administrator creates the two advertisements shows in Table A.12.

Table A.12 Advertisements

Name	Package	Program	Collection	Mandatory	Schedule	Note
Office1	Office XP	Custom (quiet)	C1	Yes	Weekly, at midnight on Fridays	Office XP installs from local distribution points, with local resilient resources. Also, the corporate policy to install Office XP on new systems is met.
Office1a	Office XP	Custom including System Files Update (quiet)	C2	Yes	Weekly, at midnight on Saturdays	Office XP installs from local distribution points, with local resilient resources. The restart occurs during non-business hours. Also, the corporate policy to install Office XP on new systems is met.

Step 5: Check the Progress of the Installation

To check the progress of your installation:

1. Monitor the installation.
2. Conduct inventory and reporting of the Office XP installations.
3. Run a query to confirm that Office XP was successfully deployed.

Step A: Monitoring the installation

After distribution starts, you can use the SMS 2003 status system to monitor the status of your distribution, packages, and advertisements. In SMS 2003, you can get detailed package status and advertisement status.

Step B: Conducting inventory and reporting of the Office XP installations

The SMS administrator uses the software inventory system to collect and report on the finalized installation of the Office XP installation.

For detailed information about using the reporting features of software inventory, flowcharts, and query support for the software inventory function, see earlier chapters in this book.

Step C: Running a query to confirm that Office XP was successfully deployed

To ensure that Office XP has been successfully deployed to your clients, the SMS administrator runs a query by mapping the Select statement to Managed Object Format properties.

Maintaining and Updating Your Office XP Installation

When Office XP is deployed in your organization, you can maintain and update it as follows:

- Distribute an Office XP public update
- Distribute an Office XP service pack
- Update Office XP installation settings

Distributing an Office XP Public Update

There are two options for the deployment of public updates:

Administrative patching This is recommended, if the instructions in the "Performing Administrative Patching of an Office XP Public Update" section are followed.

Client (standard) patching Client patching might be more appropriate in some scenarios than in other scenarios. Instructions are provided for this approach in the "Client Patching of an Office Public Update" section later in this chapter.

Performing Administrative Patching of an Office XP Public Update

When you perform administrative patching, you apply the public update patch to the Office source in the package source directory, which is then replicated to the distribution points.

You can apply patches to the original source or to a new source.

Applying a Patch to the Original Source

With this option, you apply the public update to the original package source, which is then replicated to the distribution points to update the Office source. If the clients are required to initiate a repair or installation-on-demand operation that requires the original source, then a Windows Installer error appears on the client computer. However, if the synchronization advertisement is received by most clients within a few hours of the distribution point update (which is reasonable for most SMS configurations), then it is unlikely that problems will occur on the clients. Therefore, as soon as the patch has reached all distribution points in a site, an advertisement should be distributed with the program update to the clients, so the Windows Installer source can be synchronized.

To apply a public update to the administrative source

1. Download the self-extracting file for the update from the Office Resource Kit Web site (http://www.microsoft.com/office/ork/xp/default.htm) and double-click the file name to extract the MSP file.
2. Connect to the server share for your administrative installation point (the SMS package source directory).
3. On the **Start** menu, click **Run** and then type the command line for Windows Installer with the appropriate options for your installation. Use the following syntax:

```
[start] msiexec /p [path\name of update MSP file]/a
[path\name of MSI file] SHORTFILENAMES=TRUE /qb
/L* [path\name of log file]
```

To create and distribute the program update for clients

1. Create a program for each site called Office Synchronization Program.
2. Add the command line:

```
setup.exe /fvm
```

   For setup command-line options, see http://www.microsoft.com/office/ork/xp/appndx/setopt.htm
3. Create an advertisement for each site to advertise this program

Applying a Patch to a New Source

To avoid the potential for errors on the client due to an incorrect Windows Installer source on the distribution points, an alternative method is to create a new SMS package share specifically for the newly patched Office source, and then direct clients to the share using a program advertisement. The clients that have not received the advertisement are still working from the original Windows Installer source, and all clients are always referencing a valid Windows Installer source for repair operations. This method is more complex particularly if resilient resources are being used in the transform file.

To apply a patch to a new source:

1. Create a new directory and place an administrative installation of Office XP in this directory.
2. Configure the source as described in the "Deploying Office XP" section earlier in this appendix. The new source directory should be identical to the one currently in use.
3. Apply the Office XP public update to the new source as described in the "Applying a Public Update to the Administrative Source" section earlier in this appendix.
4. Create a package with this source and distribute the package to the clients following the steps described in the deployment section earlier in this paper.

Client Patching of an Office Public Update

Client patching involves distributing the public update bundle to each client computer and running it using SMS software distribution. There are two options for distributing update files:

- Distribute the client update file that is downloaded from the Office Web site. You can do this by typing the name of the .exe file in the command line of an SMS program and including the file in the source directory for the package.
- Extract the file to obtain the raw Windows Installer patch files (.msp). You can extract the .msp files from the client update as follows

```
C:\path to update file\MyUpdate.exe /c /t:C:/folder for extracted files
```

Distributing an Office XP Service Pack

The processes for distributing Office XP service packs and public updates are similar. However, distributing a service pack provides a new baseline Windows Installer source for future Office public updates. As a result, a public update released after Office XP SP 1 can be applied only to an Office XP source that has already been patched with the Office XP SP1 update. This is an issue for client (standard) patches only.

The procedure for administrative patching of the Office source is the same as that for a public update described in the previous section.

Updating Office XP Installation Settings

To update your Office XP installation settings, you must:

1. Create updates using the Custom Maintenance Wizard.
2. Apply the .cmw file to the client.

Creating Updates Using the Custom Maintenance Wizard

After an Office XP installation has been deployed, it is not possible to change the installation settings using a Windows Installer transform file without reinstalling Office XP. However, you can update these settings dynamically on the client by using a configuration maintenance file (.cmw) produced by using the Custom Maintenance Wizard.

The Custom Maintenance Wizard is a tool that allows administrators to maintain existing Office XP installations on clients. It works with Windows Installer version 1.1 and later. Office XP Setup installs Windows Installer version 1.1.

By using the Custom Maintenance Wizard, you can modify features that you set in the Custom Installation Wizard, including default user settings, security levels, Outlook settings, and registry keys.

The Custom Maintenance Wizard creates a .cmw file based on the Windows Installer package (.msi file) used in the Office XP installation.

Applying the .cmw File to the Client

To apply the changes to a client, run the Custom Maintenance Wizard from a command line on the client specifying the customization file that contains the changes you want to apply.

As described in the *Microsoft Office XP Resource Kit*, you might want to deploy Maintwiz.exe to your clients as part of the Office XP package, so that updates made through the Custom Maintenance Wizard can be more easily deployed to clients. See the *Microsoft Office XP Resource Kit* for more information. The Custom Maintenance Wizard is automatically installed on your computer when you install the Office XP Resource Kit.

Using Resilient Sources

Depending on your environment, you can take advantage of an advanced installation option called resilient sources.

Resilient sources are created when multiple sources of Office XP (administrative installation points on a distribution point) are made available to the clients. By default, Office XP returns to the share point from which Office XP was installed. If you choose to use resilient sources, clients use those sources whenever the original distribution point is unavailable.

Creating resilient sources is valuable because:

- They provide fault tolerance.
- They enable the number of SMS distribution points with Office XP sources to be reduced after deployment.

These share points can be located on many different SMS distribution points. The resilient sources do not have to be SMS distribution points, because network share points of any type can be used to provide self-healing and sources for resilient installations of Office XP. However, over time you might choose to reduce the number of distribution points. Typically, the path to the distribution point you use for installation servers is stored by default as a source. But, if servers are unavailable, then its role as a distribution point path is no longer valid. Alternatively, administrators might want to have several sources during initial rollout but then reduce the number of sources to save disk space. Establishing resilient sources allows administrators to reduce the number of distribution points, while continuing to allow installation on first use and repairs to take place. The clients would use their list of resilient sources, and when the primary source was unavailable, they could use a second or third resilient source. If no resilient resources are available, the user is instructed to provide a path to the Office XP source files.

Windows Installer requires users to retain access to the source files throughout the life of the programs, so that advanced features, such as program repair and Install on Demand, can return to a share point other than the originating distribution point. Office XP provides the option of establishing a valid universal naming convention (UNC) path (for example, \\<*server*>\<*share*>), or a list of valid UNC paths or mapped drives for the source files.

Using the Custom Installation Wizard, you can set up a list of sources for Office XP to use when it requires source files. The resilient source list remains available for as long as Office XP is installed on the clients.

You must have the name of each server you want to use for resilient sources in order to populate a property called SOURCELIST. To configure the Windows Installer SOURCELIST parameter for the Office XP installation, use the Custom Installation Wizard. In the **Identify Additional Servers** page of the Custom Installation Wizard, enter the server names you want to use as resilient sources.

When you create resilient sources, keep in mind the following:

- In the Custom Installation Wizard, you must enter the server name and the share name. To use resilient sources, you must use a customized name in the SMS package creation process. This is located in **Share Distribution Folder** on the **Data Access** page in the SMS Administrator console. Knowing the share name in advance is necessary when creating the .mst file in the **Additional Servers** phase of the Custom Installation Wizard.
- It is important to define a share distribution folder on the **Data Access** tab of the **Package Properties** item in the SMS Administrator console. If you do not provide a name for this folder, SMS uses a default name—the package ID or the drive ID—that is not always known. For large organizations, the package ID is not predictable in advance, so it is difficult to anticipate the complete path to the distribution point share, unless it is set specifically in **Share Distribution Folder**.

- If you use resilient sources, and you have more than one site in your hierarchy, you might need to create a separate customization file (transform file) to reflect the different resilient sources. Then, use a separate SMS program within your package that points to each .mst file. You might have one for each site. Configuring separate transform files for each individual site allows each site's clients to carry a local list of resilient resources, which are most likely the distribution points for that site. This avoids unnecessary WAN traffic.

  The enumeration of the resilient sources should include the planned network share points for Office XP, whether they are SMS distribution points or not. As a result, you will have a program that is specific to each site which references the Windows Installer transform file containing the resilient sources specific to that site with all other options identical.

When you use SMS 2003 to distribute Office XP, SMS randomly selects the distribution point to which specific clients connect to for installations. This effectively distributes the load on the available distribution points. In some ways, the administrator can also control load balancing by selecting collections and distribution points for an installation. For example, if you choose to install Office XP to 10,000 users simultaneously, you will probably create stress on your network. SMS 2003 provides tools for simulating your network traffic and estimating the results of such actions.

After you install Office XP, there is additional stress to the system when installations are repaired and Install on Demand features install. You can provide more load balancing by configuring additional resilient sources for installing these components rather than just the installation distribution point.

For more information about resilient sources, see the Office XP Resource Kit Web site at http://www.microsoft.com/office/ork.

Frequently Asked Questions

How does the System Files Update relate to the installation of Office XP?

Office XP requires minimum versions of a set of .dll files and other shared and system files. Before installing Office XP on Windows NT Workstation 4.0, Windows NT Server 4.0, or Windows 98, Setup verifies that these key system files are current. If they are not, then Setup updates them automatically from the System Files Update before proceeding with the rest of the installation.

The release versions of Windows 2000 Professional and Windows 2000 Server include an equivalent or later level of the key system files required for Office XP. You cannot install the System Files Update or run Internet Explorer Setup from the System Files Update, because the **/spforce** command-line option has no effect on these operating systems.

If you are upgrading from Office 2000 Public Update 1 under Windows 98, Windows NT Server 4.0, or Windows NT Workstation 4.0, and you have Internet Explorer 5.01 or later, then your system files are also current. In these situations, you can install Office XP without installing the System Files Update.

Can I send a command line that installs the SFU to clients that already have updated system files (Windows 2000 Professional, Windows 2000 Server, Office 2000 Public Update 1)?

Yes. Sending a command line will be successful; however, the program will only detect, not install, the SFU. The Office XP installation will proceed without the SFU, because it is unnecessary to install the SFU again. A restart will occur. For example, if an organization upgrading to Office XP has systems that require the SFU and systems that do not require the SFU, then it is possible to use one advertisement, one .mst file, and one command line in the SMS program. This command line (created by the .sms file) must include the SFU portion of the command line. All systems will restart and upgrade to Office XP, including the computers that do not require the SFU. The computers that do not need the SFU will restart unnecessarily, however, this might be an acceptable tradeoff for simplicity in the right conditions.

Why is there a delay when running Office XP advertisements?

A delay occurs because SMS is designed for large organizations. For scalability reasons, it is not designed to run advertisements immediately. For example, because the SMS client is poll-based, it is important to ensure that delays are not due to a delay in the polling cycle.

Also, SMS does not run an advertisement until the package is available on the distribution point, and this can take time with large packages. You can check on the availability of the SMS package by viewing the contents of the package's share name, as specified by the administrator creating a custom share name. You can also verify the package state in the SMS Administrator console.

If the administrator specifies a custom share name for the package, the share is placed on the root of the drive with the most available space. If the administrator uses the default share name, the files are created under the Smspkg<drive letter> share. The share size depends on your configuration.

Can I suppress all screen display for Office XP distributions?

Yes. The .sms file creates packages that include different command lines. Quiet programs do not require any user interaction. However, the Windows Installer informational dialog box does appear on the screen. If the SMS administrator wants to suppress the Windows Installer screen, this can be accomplished in the command line by substituting **/qn** for **/qb-**. For complete command-line control details, see article 283686, "OFFXP: Setup Command-Line Switches," at http://support.microsoft.com?kbid=283686.

Does Microsoft provide application-specific installation and setup information for the applications within Office XP?

Yes. This information is available at http://support.microsoft.com/?kbid=293303.

How large is the administrative installation share I create for Office XP?

It is approximately 630 MB; however, this might vary, depending on your environment.

Where are the Office XP Setup log files located?

By default, Office XP log files are located in the %temp% directory on the clients. This can be redirected for reporting or file collection purposes. The log file names reflect the version installed; for example, "Office XP Professional with FrontPage setup(0001)_task(0001).txt." The name increments the file name on multiple attempts, so that "setup" is incremented.

Which edition of Office XP should I install?

The decision points for this issue are discussed in detail at http://www.microsoft.com/office/ork/xp/journ/Q&A_0001.htm.

How large is the compressed source file that SMS replicates between sites?

It is approximately 350 MB; however, this might vary in your environment.

How much disk space does the SMS site server require, if all Office XP deployment storage is located on the site server?

The administrative installation is approximately 630 MB. The package that is created is slightly smaller. The site server can also store a compressed copy for replication to other sites. This compressed copy is approximately 350 MB. There are also temp files that are associated with the compression activity. It is recommended that the SMS site server have more than 2 GB of free disk space.

What is the size of the transform files I create for Office XP?

The size varies with the specific customizations involved, but sizes between 70 KB and 1 MB have been used in some early deployments. All transform files should be included before delivering the SMS package, to minimize the amount network bandwidth that is used with a change in the package source files (and accompanying replication between sites).

Where do I put the .mst and .ops files that the Custom Installation Wizard and Office Profile Wizard install?

Both files go in the root of the administrative share before the Office XP package is created. Putting all source files in this share before creating the package ensures that when the SMS administrator creates the Office XP package, the files are only replicated once.

How can I stagger installations and decrease network bandwidth usage?

By using the Custom Installation Wizard and selecting less frequently used applications to be installed on first use, you can reduce concurrent network usage for installation of Office XP and reduce the time it takes to install Office XP.

Can I provide a highly customized Office XP deployment that configures tools, templates, and default behaviors?

The Office Profile Wizard provides this support for Office XP Setup. It creates an .ops file, which further customizes Office XP Setup. For information about the Office Profile Wizard and how to easily configure applications with it, see the *Microsoft Office XP Resource Kit.*

How do I configure Office XP to be installed locally with certain applications?

Use the Custom Installation Wizard, which sets up and creates the transform file. You can further customize the options within the application.

Where can I learn about the configuration details that are possible for each application in the Office XP suite?

For information about customizing individual applications, see the *Microsoft Office XP Resource Kit.*

Is it possible to deploy to one large collection? For example, can I use the Office XP package with the Custom with System Files Update (quiet) program, and advertise the package to the all systems collection?

Yes, this is possible. However, there are several drawbacks. First, computers that are not running Office XP-supported platforms will fail. Also, the administrator would be using the command line that is only necessary where the SFU is required on systems, such as Windows NT Workstation 4.0 and Windows 98, which are not upgrading from Office 2000 Public Update 1. It is easier to deploy all the systems together, and then target one program to run on each system. However, the SFU program incorporates a restart to allow SFU to run. This restart is unnecessary on any client that is running Windows XP Professional, Windows 2000 Server, Windows 2000 Professional, or Office 2000 Public Update 1.

What is the relationship between collections and installing Office XP?

Each collection should have a single instance of Office XP installation created by a single Office XP transform file. The .mst provides the configuration information for all Office XP installations that use the .mst to guide the installation. Also, the whole collection runs the same program, which must match the SFU and restart requirements for all the clients.

How do I upgrade Office and Internet Explorer as one package, or as a pair of linked packages?

If you want to upgrade your client's version of Internet Explorer, then you must upgrade Internet Explorer separately from your Office XP deployment. You should not chain the installation of these two products together. You can deploy Internet Explorer either before or after you deploy Office XP. Office XP automatically installs Internet Explorer 5.01 during SFU. Installing Office XP does not downgrade Internet Explorer versions that are later than 5.01.

Is Windows Installer required before the installation of Office XP?

No. However, Office XP operation requires Windows Installer 1.1, which is automatically installed during the SFU. Windows Installer 1.1 contains a number of improvements to Windows Installer 1.0, including better upgrade support.

For more information about Windows Installer and SMS, see the following documents:

- http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/sms/deploy/depovg/deplymsi.asp?frame=true.
- http://www.microsoft.com/windows2000/techinfo/administration/management/wininstaller.asp.

Where can I get the Office Resource Kit tools?

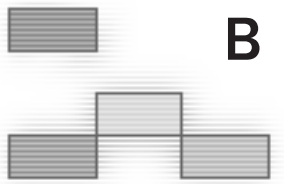
You can download the Custom Installation Wizard from the Toolbox in the Office XP Resource Kit at http://www.microsoft.com/office/ork/xp/default.htm

Can I deploy to clients that are running Terminal Services?

You can use SMS 2003 to deploy Office XP to clients that are running Windows 2000 Terminal Services. For additional information about using Terminal Services with SMS, see the "Using SMS 1.2 and SMS 2.0 with Windows 2000" white paper at http://www.microsoft.com/smsmgmt/deployment/wincompat.asp.

APPENDIX B

Windows Management Instrumentation



Microsoft® Systems Management Server (SMS) 2003 is a Windows®-based product that you can use to manage, support, and maintain a distributed network of computer resources. SMS enhances its abilities by using Windows Management Instrumentation (WMI), which is management infrastructure that supports monitoring and controlling system resources through a common set of interfaces.

You can use SMS successfully without understanding WMI. However, if you want to extend or automate SMS, or troubleshoot certain types of SMS problems, you need to know how to use WMI and understand its relationship with SMS.

In This Appendix

- Introduction to WMI
- How SMS Uses WMI
- Looking at WMI
- Managing WMI
- Troubleshooting WMI
- Learning More About WMI

Introduction to WMI

WMI is a middle-layer technology that enables standardized management of Windows-based computers. It collects computer management data from a wide variety of sources and makes it accessible by using standard interfaces. WMI can be accessed remotely, but it does not consolidate the management data in a central location — that is one of the functions of SMS. You can also use WMI to set configuration details on your computer and to detect and respond to changes in the configuration of your computer (using WMI events).

WMI is the Microsoft implementation of the Web-based Enterprise Management (WBEM) standard, which was developed by the Desktop Management Task Force.

WBEM builds on a series of industry-wide initiatives to standardize computer management. Those initiatives include the Simple Network Management Protocol (SNMP) and Desktop Management Interface standards. Standardizing computer management allows you to use multiple tools to address your various computer management needs without requiring a separate infrastructure for each tool. For example, SMS and Microsoft Operations Manager (MOM) both use WMI. You can use SMS for centralized inventory collection of all your computers and use MOM to alert you to critical changes in your servers, but your servers do not require two different types of client agents to collect configuration information. Using WMI you can also write scripts to change computer settings and then distribute those scripts to your SMS clients.

WMI is available for all Microsoft Windows® 95 or later operating systems. It is installed with Microsoft Windows Millennium Edition, Windows 2000, Windows XP, and the Microsoft Windows Server™ 2003 family and is available for download for the other supported versions of Windows. There are three significant versions of WMI:

- Version 1.1 (build 698), which ships with SMS 2.0. It is also included with Microsoft Windows 98 Second Edition and Windows NT 4.0 Service Pack 4 and later.
- Version 1.5 (build 1085), which ships with SMS 2003 and Windows 2000. It is available for all WMI-supported operating systems except Windows XP and the Windows Server 2003 family.
- Version 5.1.2600.0, which ships with Windows XP and the Windows Server 2003 family.

You can determine which version of WMI is installed on your computer by using the WMI Control. For more information, see the “Using WMI Management Tools” section later in this appendix.

WMI can collect and set configuration details for a wide variety of hardware types, operating system components and subsystems, and application systems because it uses *providers* to work with those systems. Providers are relatively simple components that the developers of the systems make available. WMI uses the providers to interact with the systems. Any system that has a WMI provider can be managed with WMI and can therefore be managed by any computer management application (such as SMS and MOM) that uses WMI. WMI providers are available for a wide variety of systems including:

- Microsoft Exchange
- Microsoft SQL Server™
- SNMP
- Domain Name Service (DNS)
- Active Directory® directory services
- SMS
- The operating system
- Windows Installer
- Microsoft Internet Explorer
- Microsoft Office 2000 and Office XP
- Internet Information Server (IIS)
- Cluster
- Systems Network Architecture (SNA)
- Windows drivers (WDM)
- HTTP Monitoring
- COM+ Monitoring

Many other providers are also available, and a variety of computer system vendors are developing additional WMI providers. For more information about WMI support, see your system documentation. WMI support is usually included when you install the system, but in some cases there might be extra setup steps that you must follow.

Using WMI, WMI-based management applications, and the WMI providers included with your systems, you can efficiently manage all your computers and their systems with a small set of management tools.

How SMS Uses WMI

SMS uses WMI to collect information about SMS clients and for the management and operation of SMS itself. Specifically, SMS uses WMI for:

- The SMS Administrator console, which gathers and sets all SMS configuration and inventory details (using the root\SMS\site_<*sitecode*> namespace).
- Resource Explorer, which gathers inventory data from the same namespace as the SMS Administrator console, but uses the group classes for a specific computer. Resource Explorer also uses the **SMS_PropertyDisplayNode** class in the root\SMS\inv_schema namespace for window formatting details.
- All other SMS administration tools, including support tools, resource kit tools, recovery tools, scripts, and any tools you have written with the SMS software development kit (SDK).
- The Legacy Client Hardware Inventory Agent, which gathers hardware inventory data from WMI (by default from the root\CIMv2 namespace), and records which WMI classes should be reported by the Legacy Client Hardware Inventory Agent. The classes that the agent should report are recorded in the root\CIMv2\SMS namespace. The settings in that namespace are usually transferred to clients by using the SMS_def.mof file. The \root\CIMv2\SMS\Delta namespace records the values of the classes when they were last reported, so that hardware inventory deltas can be calculated.
- The Advanced Client. Configuration policies and most client data are stored in WMI in the root\CCM namespace and its namespaces.
- The Advanced Client Inventory Agent, which gathers hardware inventory data from WMI (by default from the root\CIMv2 namespace) and records the values of the classes when they were last reported, so that hardware inventory deltas can be calculated.
- The SMS Provider and SMS Component Management Provider, which use WMI as the interface model and store some details in the WMI repository. The SMS Provider also uses the root\SMS\inv_schema namespace for localization details.
- Queries and collections, which use the WBEM Query Language (WQL) and the SMS Provider.
- Network Discovery. Configuration options are stored in the root\NetworkModel namespace.
- Network Trace, which uses the SMS Component Management Provider and SMS Component Polling Provider.
- Reporting (using tools other than the reporting tool included in SMS 2003 or that use the SMS SQL Server views), which uses the WBEM Open Database Connectivity (ODBC) driver. ODBC is a commonly used database interface.
- A pointer to the SMS Provider server, which uses the **root\SMS\SMS_ProviderLocation** class.

Because SMS uses WMI, you can:

- Script SMS operations to ease your SMS administration tasks. For more information, see Appendix C, "Scripting SMS Operations."
- Increase or decrease the details that SMS collects as part of the hardware inventory. For more information, see Chapter 2, "Collecting Hardware and Software Inventory."
- Build tools to manage SMS. For more information, see the SMS SDK.
- Directly connect to client computers, if they are accessible on your network, to verify in real-time any details that you see in Resource Explorer. For more information, see the "WMI Browsing Tools" section later in this appendix.
- Report, query, and build collections based on any configuration details that are available from client computers. For more information, see Chapter 4, "Managing Collections and Queries," and Chapter 11, "Creating Reports."

Understanding WMI

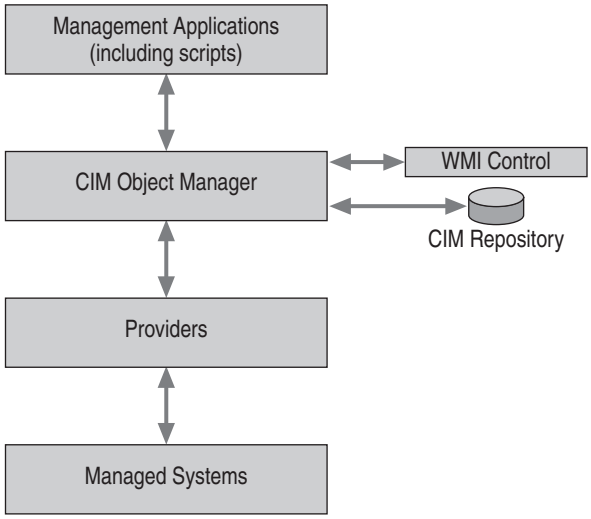
A good way to become comfortable with WMI is to look at it from the various perspectives that can be important to an administrator:

- Architecture — how it is implemented and how it works
- Object model — how programs and scripts access data
- Schema — how data that WMI manages is organized
- Tools — how you can work with WMI

WMI Architecture

WMI is designed to function as a middle layer, by serving as a standard interface between management applications and the systems that they manage. Figure B.1 illustrates the WMI architecture.

Figure B.1 WMI architecture



The Common Information Model (CIM) Object Manager is implemented in the WMI service on Windows NT 4.0, Windows 2000, Windows XP, and operating systems in the Windows Server 2003 family. The WMI service runs WinMgmt.exe and related dynamic-link libraries (DLLs) from the %*Windir*%\System32\Wbem directory. In Windows XP and operating systems in the Windows Server 2003 family, the WMI service runs as part of a Svchost service.

Using WMI Control, you can back up the repository, control logging, and configure elements of WMI.

The CIM Repository is a database for static WMI data and object definitions. The CIM Repository is stored in the CIM.rep file in the %*Windir*%\System32\Wbem\Repository directory or in files in the Repository\FS directory for Windows XP and later operating systems. The CIM Repository is built during WMI setup from Managed Object Format (MOF) files, such as CIMwin32.mof, that are found in the %*Windir*%\System32\Wbem directory.

With the exception of Advanced Client settings, the CIM Repository does not contain data. It mostly contains definitions. Most of the data is retrieved dynamically and is therefore only accurate at the time that you ask for it. The classes that are defined to use providers specify which provider to use and supply sufficient details for the provider to get the data. When WMI receives a request for data, it checks the class definition for the details and then asks the relevant provider to get the data from the system.

Other possible directories in the %*Windir*%\System32\Wbem directory tree are:

- AdStatus — for the trust monitor provider.
- Logs — see the "WMI Troubleshooting Techniques" section later in this appendix.
- Mof — MOF files placed in this directory are automatically compiled. Those that are successfully compiled are placed in a subdirectory named Good. Those that fail to compile are placed in a subdirectory named Bad. Using this directory is discouraged.

- Performance — for the high-performance provider.
- Snmp — for the SNMP provider.
- Xml — for files to support XML extensions to WMI.

Providers can be implemented as separate DLLs, in DLLs with other functions, or in an application executable file. They can operate in the context of the WMI service or the application. Providers can be located in any directory. If installed, the WMI SDK files can be found in the \Program Files\WMI directory.

Common WMI file types and file name extensions are:

- .mof — for Managed Object Format files. For more information, see the "Using MOF Files" section later in this appendix.
- .mfl — for MOF localization files.

You can connect to WMI on any computer on your network that has WMI installed, if the security configuration on that computer allows it. You can remotely connect to WMI by using the Distributed Common Object Model (DCOM) protocol, which uses the remote procedure call (RPC) protocol, which in turn uses TCP/IP.

The WMI registry tree is **HKLM\Software\Microsoft\WBEM**.

Internal components of WMI include:

- The event subsystem — processes WMI events.
- The Framework — the core providers including the Microsoft Win32 provider.
- Connection proxy — facilitates connections between WMI-based applications and WMI using DCOM.

For computers running Windows XP or operating systems in the Windows Server 2003 family, WMI providers are loaded into one or more processes running WMIPrvse.exe. The providers provide the same functionality as before, but they do it in a more secure and reliable manner.

WMI Object Model

Management applications and scripts work with WMI through the WMI Object Model, as illustrated in Figure B.2. The object model defines the programming interface to WMI. The WMI Object Model illustrates how different elements within it relate to each other. It also shows the relationship between the elements, not the flow of data.

Figure B.2 Simplified WMI Object Model relationships between the WMI locator, service, properties, methods, qualifiers, and other objects

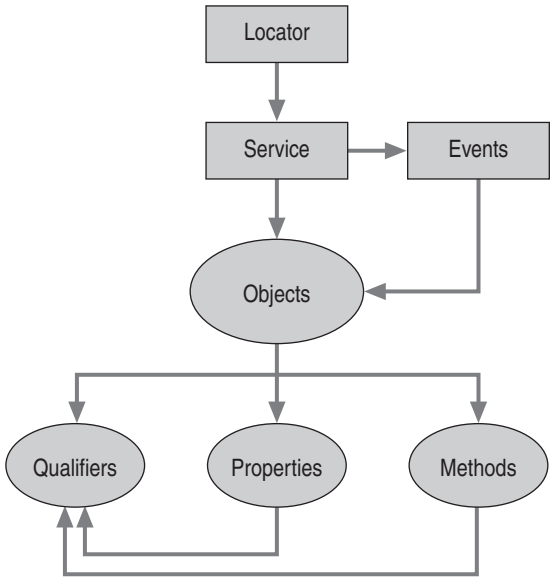


Figure B.2 is simplified in that it does not include all the elements of the WMI Object Model, such as object paths, named value sets, event sinks, and other less commonly used elements. Scripts commonly use the elements illustrated in Figure B.3. Other elements of the WMI Object Model can be used in scripts but they are all used in the context of the elements illustrated. For more information about the WMI Object Model, see the WMI SDK.

The elements of this simplified WMI Object Model are:

Locator Used to connect to the WMI service on a computer.

Service object Used to connect to the WMI service on a computer and is the main point of contact to WMI for programs.

Objects Fundamental representations of computer elements and are used by WMI and your scripts to identify to providers which specific elements that you want manipulated.

Events Changes to WMI objects. Events can be captured as objects and then manipulated in the same ways that any other objects, except that they cannot be changed or saved in WMI.

Properties Supplies descriptive or operational information about an object. For example, a **Win32_DiskDrive** object includes a property called **InterfaceType**, which might have the value of IDE for your C: drive. Properties can also be set to particular values, if the property is changeable. Setting **InterfaceType** to SCSI is not appropriate, because the only way to change the actual interface type is to replace the controller card. However, you can set a share name to a different value.

Methods Actions that you can execute on objects. For example, a **Win32_Directory** object includes a method called **Compress()** that allows the contents of a folder to be compressed in the same way as can be done by using the Windows graphical user interface.

Qualifiers Characteristics of objects, properties, and methods. For example, a qualifier for a property might indicate that it is read-only, or it might list the allowable values for the property. A qualifier for an object might be that it is read-only.

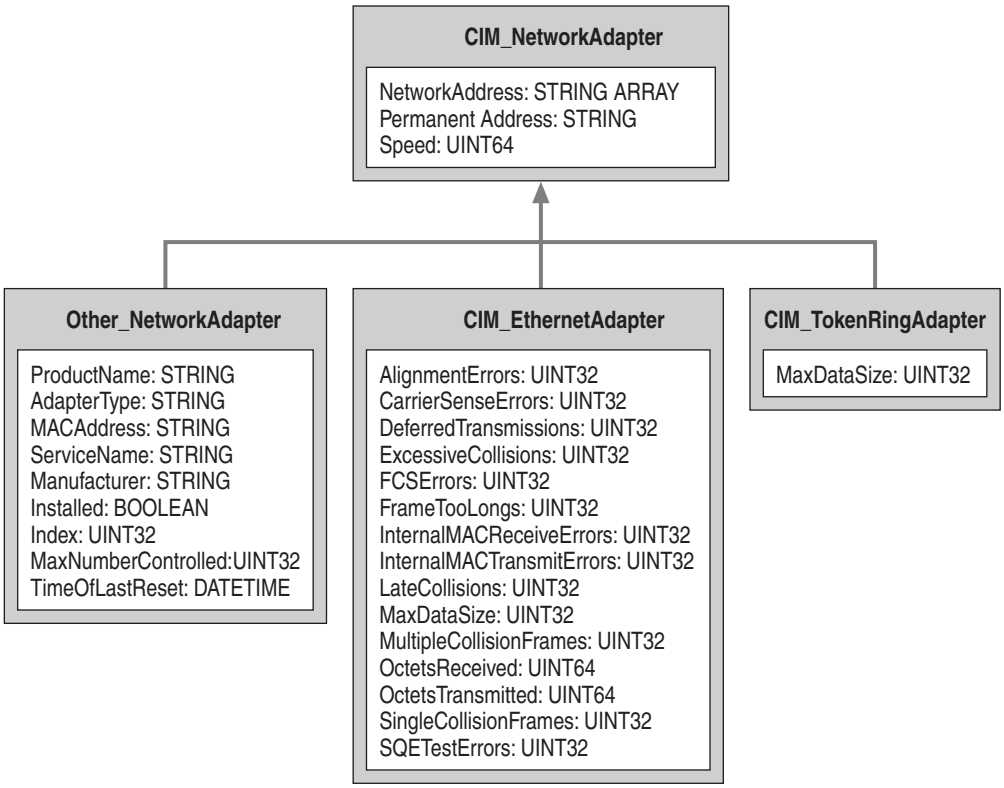
For examples of scripts that use the WMI Object Model to access SMS data, see Appendix C, "Scripting SMS Operations."

WMI Schemas

While 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, whereas a map illustrates where the destinations are and how to get to them. The driving manual is analogous to the object model, while maps are analogous to schemas. Understanding WMI schemas allows you to understand the relationships among the objects that WMI manages.

Part of a WMI schema is illustrated in Figure B.3. In this case, specific types of network adapters are defined by extending a general definition of network adapters (CIM_NetworkAdapter).

Figure B.3 Part of a WMI schema



WMI objects are described by classes, providing definitions of their properties, attributes, and other information. These classes are organized into an inheritance hierarchy supporting object associations and grouped by areas of interest, such as networking, applications, and systems. Each area of interest represents a schema, which is a subset of the information that is available about the managed environment.

The Desktop Management Task Force 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.

Some important concepts to understand about WMI schemas are:

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

Class A definition of objects. Classes define the properties, their data types, methods, associations, and qualifiers for both the properties and the class as a whole.

Instance 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..

MOF Managed Object Format.

MOF file A definition of namespaces, classes, instances, or providers; but in a text file. For more information, see the "Using MOF Files" section later in this appendix.

MOF compiling Parsing a MOF file and storing the details in the WMI repository.

CIM Common Information Model. For more information, see the Desktop Management Task Force Web site at http://www.dmtf.org/standards/standard_cim.php.

Association A WMI-managed relationship between two or more WMI objects.

You can extend a schema by adding new classes and properties that are not currently provided by the schema. For information about extending the WMI schema, see the WMI Tutorial at http://www.microsoft.com/downloads/release.asp? releaseID=12570.

Comparing WMI to SQL Server

Another way to better understand some key WMI elements is to compare them with SQL Server elements. You might have worked with a database system, such as SQL Server, and understand those concepts.

SQL Server and WMI are radically different systems serving very different purposes. WMI can store some data, but it is not built for the efficient large-scale storage and retrieval of data for which SQL Server is built. The programming interfaces are also completely different. However, relating some SQL Server concepts to roughly equivalent WMI concepts can help you to better understand WMI. Table B.1 lists some analogous concepts in the two systems.

Table B.1 Analogous Concepts Between SQL Server and WMI

Conceptual element	SQL Server	WMI
Individual items	Rows	Instances
The characteristics of items	Columns	Properties
Containers of columns and rows	Tables	Classes
Container of tables	Databases	Namespaces
Program code that functions on data	Stored Procedures	Methods
Table characteristics	Table characteristics	Class Qualifiers

WMI Browsing Tools

Browsing the WMI classes and instances is another effective method that you can use to become comfortable with WMI. Reasons why you might have to directly work with WMI include:

- An SMS component that uses WMI is not working. When you understand how the component is supposed to work, you can confirm the existence or flow of data in WMI to help you isolate the problem.
- WMI does not seem to be working properly. You can confirm that the usual namespaces are available, that they have a full complement of classes, and that you can enumerate important instances.
- You want to write queries, reports, or scripts or extend the inventory collection. Looking at the data as it exists in WMI helps you to confirm that the data you require is available and confirm which classes, instances, and properties you must reference.
- You want to adjust the way that Resource Explorer displays nodes, as described in the SMS SDK.

Several tools are available that allow you to browse or manipulate WMI data:

- CIM Studio
- WBEMTest.exe
- Microsoft Visual Studio® .NET
- WMI Command-line (WMIC) tool
- Other WMI browsing tools

These tools can also be used to manipulate WMI data, execute methods and queries, and perform other WMI functions. Browsing is just the most common use.

Caution

The browsing tools can be used to delete or modify classes and instances. However, deleting or modifying SMS classes and instances can result in the loss of valuable data and cause SMS to function unpredictably. Deleting SMS classes can cause tools that use the SMS Provider, such as the SMS Administrator console, to not work.

CIM Studio

The easiest tool for browsing WMI is CIM Studio. CIM Studio is available as part of the WMI tools, which are available from the Microsoft Web site at http://msdn.microsoft.com/downloads/.

After you install the WMI SDK, you can start CIM Studio from the WMI SDK program group. CIM Studio prompts you for the name of a namespace. If you are not sure which namespace to use, you can click **Browse for Namespace**. An explanation of how SMS uses WMI namespaces is listed in the "How SMS Uses WMI" section earlier in this appendix. In the **Browse for Namespace** dialog box, click **Connect**, and then click **OK**.

After selecting a namespace, a window appears. The left pane is the class explorer, which you can use to browse the class names. The right pane is the class viewer, which shows the properties of the currently selected class. You can also use the **Methods** tab to display the methods that are available for the class.

Table B.2 lists the commonly used buttons and icons for WMI CIM Studio.

Table B.2 Commonly Used CIM Studio Buttons

Button	Function
	Search for Class — most classes are intuitively named, so you can search for a class that includes what you are looking for in the name. For example, if you need a class that gives memory details, search for the word memory.
	Browse for Namespace — if you decide that this namespace is not what you need, you can go back and browse for a more appropriate namespace.
	Instances — if the class looks promising and you want to see some actual data, you can use this button to see instances of the class.
	WQL Queries — if the class has many instances, or if you only want to see particular instances, you can use this button to run a query.
	Help for class — many classes include descriptive text that describes the purpose of the class and its properties and methods. The details provided for the methods include any parameters that the methods can use.
	Help — full details about CIM Studio. Note that it must be double-clicked.

CIM Studio also includes the MOF Generator Wizard, which can save the definition of WMI objects as MOF files. If the objects are not computer-specific, the MOF files that are created by the MOF Generator Wizard can be transferred to another computer and compiled there to make the objects available on that computer as well.

WBEMTest.exe

WBEMTest.exe is included on every computer that has WMI installed. You can use it to quickly explore or confirm WMI details. However, WBEMTest.exe is only designed to be a support tool and has little support for browsing classes or instances.

On computers running Windows XP and operating systems in the Windows Server 2003 family, Help is available for WBEMTest.exe by clicking **Help**.

Visual Studio .NET

Server Explorer in Visual Studio .NET can be extended to use and display WMI data. Both classes and instances are displayed. Properties for the currently selected class or instance are displayed in the Visual Studio .NET **Properties** dialog box. You can run methods by right-clicking the item in the **Management Classes** tree and selecting the method from the menu.

You can download the management (WMI) extensions for Server Explorer from the Microsoft Web Site at http://www.microsoft.com/downloads/release.asp?ReleaseID=31155.

WMI Command-line Tool

The WMI Command-line (WMIC) tool provides a simple command-line interface to WMI. This allows you to use WMI to manage computers running Microsoft Windows. You can use WMIC from any computer running Windows XP Professional or an operating system in the Windows Server 2003 family to remotely manage any computer with WMI installed. WMIC does not have to be available on the remotely managed computer for WMIC to manage it.

WMIC allows you to:

- Browse the WMI schemas and query their classes and instances, usually by using aliases that make WMI more intuitive.
- Work with the local computer, remote computers, or multiple computers by using a single command.
- Customize aliases and output formats to suit your needs.
- Create and execute scripts that are based on WMIC.

The WMI infrastructure is accessible as you use WMIC through intermediate facilitators called *aliases*. Aliases are used to capture the features of a WMI class that are relevant to some specific task, such as disk or network administration. Aliases can be used to provide better names for WMI classes, properties, and methods or to arrange properties in useful output formats. The output formats can include specific property values or be formatted in a manner that is appropriate to some specific presentation strategy or function. For example, an alias might have a BRIEF format that lists only property values that are essential for the identification of the objects visible through the alias. Management data is retrieved in XML format and processed by built-in or custom XSL output formats.

To start WMIC in interactive mode

1. On the taskbar, click **Start**, and then click **Run**.
2. In the **Run** dialog box, type **WMIC**, and then click **OK**.

   The following appears, where **root\cli** is the default WMIC role: **wmic:root\cli**.
3. At the command prompt, enter an alias, command, or global switch, or enter **/?** for Help.
4. When you are done with WMIC in interactive mode, type **Exit** or **Quit**, and then press ENTER.

WMIC includes Help at the command line. At any level you can type **/?** and get additional details. By itself, **/?** provides the available global switches and the aliases that are available in the current role. When used after an alias, **/?** provides the verbs and switches available for that alias. After a verb, **/?** provides the details for that verb.

For example:

```
wmic:root\cli /?
```

Provides a list of the syntax and available aliases, including the process alias.

```
wmic:root\cli process /?
```

Displays options that are available for the process alias.

For more information about WMIC, see the Windows XP Help or the Help in the Windows Server 2003 family.

To use WMIC with SMS, try commands such as:

```
wmic:root\cli /namespace:\\root\SMS\site_MSO
wmic:root\cli PATH SMS_Collection
wmic:root\cli PATH SMS_R_System.LastLogonUserName='PTHOMSEN'
wmic:root\cli /namespace:\\root\cimv2
```

The last line is necessary to return WMIC to its normal namespace, as used in the predefined WMIC aliases.

Other WMI Browsing Tools

You can use other tools to look at WMI. If you want to use any ODBC-compatible application (such as Microsoft Excel or Microsoft Access), you can use the WBEM ODBC driver. This driver is used in the same way as any other ODBC driver, but it allows you to access information through WMI.

To save MOF data that you find while browsing (possibly for use on another computer), you can use Wbemdump.exe. Wbemdump.exe is a command-line oriented program and is included in the WMI SDK. For more information, use Wbemdump.exe with the **/?** switch.

Managing WMI

You might want to manage WMI to:

- Manage WMI setup and upgrade.
- Use WMI management tools.
- Back up WMI data.
- Control WMI security.
- Use MOF files.

Managing WMI Setup and Upgrade

WMI is included in Windows 2000, Windows XP, the Windows Server 2003 family, and Windows Millennium Edition. WMI must be installed separately for Windows 95, Windows 98, and Windows NT 4.0. SMS automatically installs WMI on SMS clients if they require it. If you want to install or upgrade WMI, see the Microsoft Web site at http://www.microsoft.com.

WMI might be upgraded with the upgrade of the operating system or the installation or upgrade of a management application, such as SMS. Verify that the systems using WMI (including SMS) are compatible with the new version prior to upgrading WMI.

If you have modified WMI objects that are defined by WMI, upgrading WMI might overwrite your changes. Therefore, you should avoid changing WMI-provided objects. If you do make such changes, record them so that you can reproduce them after the upgrade.

WMI does not have separate service packs. WMI hotfixes are sometimes included in operating system service packs. WMI hotfixes might be available separately, but hotfixes are only available to fix specific problems that are reported to Microsoft Product Support Services.

Using WMI Management Tools

You can manage WMI by using the following tools:

- WMI Control
- MOF Compiler (MOFComp.exe)
- WinMgmt.exe

WMI Control is the primary tool for managing WMI. It can be used to control WMI logging, execute WMI backups and restores, and control WMI security. On computers with WMI 1.5 or later (including computers running Windows 2000, Windows XP, or the Windows Server 2003 family), you access WMI Control by opening the Computer Management console.

To access WMI Control

1. On the taskbar, click **Start**, and then click **Control Panel**.
2. Double-click the **Administrative Tools** icon, and then double-click **Computer Management**.
3. In the Computer Management console, expand **Services and Applications**, right-click **WMI Control**, and then click **Properties**.

The Computer Management console can also be opened by right-clicking the **My Computer** icon and clicking **Manage**.

On computers with WMI earlier than version 1.5 (possibly including computers running Windows 95, Windows 98, and Windows NT 4.0), WMI Control can be found in Control Panel (or you can run Wbemcntl.exe from the %*Windir*%\System32\wbem directory). This version of WMI Control does not manage WMI security. You must manage WMI security by running Wbemperm.exe from the %*Windir*%\System32\wbem directory.

To make the contents of a MOF file effective (by placing them in the CIM Repository), the file must be compiled. MOF files are usually automatically compiled during the installation of the systems with which they are provided, but you can also compile MOF files by using the MOF Compiler (Mofcomp.exe). The MOF Compiler is available in the %*Windir*%\System32\wbem directory. You must specify the MOF file as the parameter of the MOF Compiler. You can also specify an Autorecover switch if you want the MOF file to be automatically recompiled if the CIM Repository ever has to be automatically recovered. For more information, type **Mofcomp /?** at the command prompt.

Another tool that you can use to manage WMI is Winmgmt.exe. This tool is located in the %*Windir*%\System32\wbem directory. For a list of the available switches, type **WinMgmt /?** at the command prompt.

Table B.3 WinMgmt.exe Switches

Switch	Description	Comment
/kill	Causes all instances of WMI to stop.	Use NET START "Windows Management Instrumentation" to restart WMI, or restart the computer.
/regserver	Invokes self-registration.	Only needed if the WMI service registry entries are corrupted.
/unregserver	Removes the registry entries.	Only needed if the WMI service registry entries are corrupted.
/backup <file name>	Backs up the repository. A file must be specified.	If you do not specify a path for the file, it is put in the %*Windir*%\System32 directory.
/restore <file name> <flag>	Restores the repository. A file must be specified. The flag must be 1 to disconnect users prior to the restore, or 0 to restore only if no users are connected.	
/resyncperf <WMI PID>	Registers the computer's performance libraries with WMI. WMI PID is the process ID for the WMI service.	Only needed if the performance monitor classes are not returning reliable results.
/clearadap	Clears prior /resycperf information from the registry.	Only needed if the performance monitor classes are not returning reliable results.

Backing Up WMI Data

You can back up the WMI CIM Repository to ensure that it can be restored in case of data corruption or loss. You can back up the CIM Repository by using WMI Control or by using **Winmgmt /backup** at the command prompt. You can restore the CIM Repository if you think that the WMI data is corrupted or lost. However, any changes that were made since the backup was created will be lost.

Usually, WMI has more than 600 classes. Using the scripts in the "Verifying the State of the CIM Repository" section later in this appendix, you can determine how many classes are in your CIM Repository. If you have significantly fewer than 600 classes, then a restore or recovery of the repository might be appropriate.

If you do not have a recent backup of the CIM Repository to restore, you should avoid deleting it. While it is true that WMI can automatically recover much of the CIM Repository, you will lose instances and customizations that are not included in the automatically recovered MOF files. Because you might have several management applications or operating system subsystems that rely on WMI, deleting the CIM Repository might fix errors for one but cause problems for another.

If the CIM Repository does not automatically recover when it should (for example, when the CIM Repository becomes corrupted and WMI restarts), you can try to force an automatic recovery of the CIM Repository by using the command **Regsvr32 wbemupgd.dll** (from the %*Windir*%\System32\Wbem directory). If this does not work, you can also try the command **Rundll wbemupgd.dll, RUNDLLENTRY**. These commands should force WMI to automatically recompile the MOF files that were compiled with the Autorecover option.

Understanding WMI Security

WMI benefits from the operating system's security in two ways:

- The operating system security can prevent unauthorized access to WMI.
- Anything that you can access from WMI can only be used directly from the operating system. For example, when you run a script to create shares through WMI, you can only create shares that you are authorized to create through Windows Explorer.

WMI also has its own layer of security. Prior to WMI version 1.5, which shipped with Windows 2000, WMI security only controlled who could access and manage WMI. Anyone that could access or manage part of WMI could access all of it. With version 1.5 and later, WMI security is controlled at the namespace level. For example, you can allow some administrators to obtain SQL Server details from WMI by using the SQL Server namespace and allow other administrators obtain Microsoft Exchange details by using the Exchange namespace.

WMI security is controlled by using WMI Control or Wbemperm.exe, as described in the "Using WMI Management Tools" section earlier in this appendix. Users with administrator privileges on the computer do not have to be added to WMI security — they always have full access to WMI.

Using MOF Files

Whenever you manage WMI, or systems that use WMI, you might use or see MOF files. WMI data (such as definitions of namespaces, classes, instances, or providers) are sometimes represented in MOF files. MOF files are a convenient way to change WMI settings and to transfer WMI objects between computers. The contents of MOF files only become effective when the files are compiled. For more information about compiling MOF files, see the "Using WMI Management Tools" section earlier in this appendix.

One of the most powerful uses of MOF files with SMS is to extend SMS inventory collection. This is typically done by editing the SMS_def.mof file. For more information, see Chapter 2, "Collecting Hardware and Software Inventory."

The WMI SDK provides complete details about the syntax of MOF files. The following MOF example illustrates a typical (though simple) MOF file. The highlights are:

- Lines between braces ({ and }) are part of a single definition. Some WMI elements can be defined in a single line (properties, for example), but others that require multiple lines for a definition (classes, for example) are defined within braces. The name and type of the definition immediately precedes the first brace.
- Lines between brackets ([and]) are qualifiers that apply to the definition that follows it. Qualifiers can apply to the entire class or instance, or they can apply to particular properties. The placement of the qualifier determines which level it affects.
- The #pragma lines are instructions to the MOF Compiler. Usually, these lines specify which namespace should be used. That namespace will be used for all definitions until the next #pragma line.
- If a provider is required by a class or instance definition, it must be defined in the MOF file prior to the definition of the class or instance. The details for defining the provider will be included with the documentation that is provided with the provider (or the product that the provider is part of).

```
#pragma namespace("\\\\.\\root\\cimv2")
// Instance provider
instance of __Win32Provider as $InstProv
{
    Name    = "RegProv" ;
    ClsId   = "{fe9af5c0-d3b6-11ce-a5b6-00aa00680c3f}" ;
};

instance of __InstanceProviderRegistration
{
    Provider = $InstProv;
    SupportsPut = TRUE;
    SupportsGet = TRUE;
    SupportsDelete = FALSE;
    SupportsEnumeration = TRUE;
};

[dynamic, provider("RegProv"),
 ClassContext("local|HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows NT\\
 CurrentVersion\\Hotfix")
]
class HotFixes
```

(continued)

(continued)

```
{
  [key]
  string  QNumber;

  [PropertyContext("Installed")]
  uint32 Installed;
};

#pragma namespace("\\\\.\\root\\cimv2\\sms")
[SMS_Report(TRUE),
SMS_Group_Name("Hotfixes"),
SMS_Class_ID("MICROSOFT|HOTFIXES|1.0")]
class HotFixes : SMS_Class_Template
{
  [SMS_Report(TRUE),key]
  string  QNumber;

  [SMS_Report(TRUE)]
  uint32 Installed;
};
```

Troubleshooting WMI

Problems with WMI fit into the following categories:

- Installation
- Connectivity
- Resource consumption
- Programming (usage)

By using some common WMI troubleshooting techniques and verifying the state of WMI, you can put any WMI problems you have in one of these categories. The following sections help you to do that and discuss common problems and solutions.

WMI Troubleshooting Techniques

You can troubleshoot WMI by using the following steps:

1. If the management application or script seems to be partially successful by using WMI or it returns a complex error message, see the “Programming Issues” section later in this appendix.

2. If the management application or script is running successfully but very slowly or it is causing WMI to consume excessive resources on the computer, see the "Resource Consumption Issues" section later in this appendix.
3. Ensure that WMI is installed. Verify that the %*Windir*%\System32\WBEM directory exists and that it contains WinMgmt.exe. For more information, see the "Installation Issues" section later in this appendix.
4. Connect to WMI by using a WMI browsing tool. If you cannot connect to WMI, see the "Connectivity Issues" section later in this appendix.
5. Review WMI security by using WMI Control (or Wbemperm.exe for earlier versions of WMI). Ensure that the accounts that are used by your management application are able to access WMI and the namespaces that are used by that application.
6. Use a WMI browsing tool to browse the namespace that is used by your management application. You should be able to find the relevant classes and see instances of data that should be available. If classes are missing, you might have to recompile the MOF files for the management application or reinstall the management application. If instances are missing, troubleshoot the provider for the system that is being managed (the Exchange Provider, for example) and the system itself.
7. Review the logs that are available in the %*Windir*%\System32\wbem\logs directory. The logs are described in Table B.4.
8. Use WMI Control to enable verbose logging. Reproduce the problem and review the logs again. Set WMI logging back to normal after you complete this step.
9. If the problem might be provider-specific, enable provider logging. In addition to default WMI logging, you can enable provider logging by changing the **Logging** or **Level** values in the following registry tree:

   HKLM\Software\Microsoft\WBEM\Providers\Logging
10. If you think that the CIM Repository might be corrupted, verify that it has a normal complement of namespaces, providers, and classes. For more information, see the "Verifying the State of the CIM Repository" section later in this appendix. If this confirms that the CIM Repository is corrupted, see the "Backing Up WMI Data" section earlier in this appendix.
11. For additional solutions to your problem, see the Microsoft Knowledge Base at http://support.microsoft.com.
12. If a later version of WMI, DCOM, or the operating system (even a service pack) is available, apply that upgrade.
13. Contact Microsoft Product Support Services.

Table B.4 WMI Logs

Log	Content
Wbemcore.log	Core
WinMgmt.log	The WMI service
Framework.log	The WMI framework, including the Win32 Provider
Wbemess.log	The event subsystem
Setup.log	Setup and upgrade
Mofcomp.log	MOF Compiler messages
Wbemprox.log	The connection proxy, which is relevant during the connection to WMI, especially for DCOM-related issues
WMIAdap.log	The high-speed performance counter adapter
WMIProv.log	Providers-related issues, but not necessarily provider-specific issues
WMIC.log	The WMI Command-line tool
Dsprovider.log	The Active Directory provider
NTEvt.log	The Windows NT event log provider
WbemSnmp.log	The SNMP Provider

Note

The WMI logs might have a file name extension of .lo_. This is the previous version of the log. When the logs grow to a specific size, the log is renamed with the .lo_ file name extension and a new log is created for new entries. Any previous log with the .lo_ file name extension is overwritten.

Verifying the State of the CIM Repository

If you think that the CIM Repository might be corrupted, you can verify its state by running the following scripts on the computer.

The following script lists all the namespaces:

```
'Namespaces.vbs
Set loc = CreateObject("WbemScripting.SWbemLocator")
wscript.echo "root"
GetNamespaces "root", 1
```

(continued)

(continued)

```
Sub GetNamespaces(Path, Level)
    Set WbemServices = loc.ConnectServer( , path )
    Set Namespaces = WbemServices.ExecQuery("Select * From __Namespace")
    For Each Namespace in Namespaces
        Wscript.Echo Space(5*level) & Namespace.Name
        GetNamespaces path & "\" & Namespace.Name, Level+1
    Next
End Sub
```

The following script lists all the providers:

```
'Providers.vbs
Set shell = CreateObject("WScript.Shell")
Set loc = CreateObject("WbemScripting.SWbemLocator")
GetNamespaces "root", 1

Sub GetNamespaces(Path, Level)
    Set WbemServices = loc.ConnectServer( , path )

    Set Namespaces = WbemServices.ExecQuery("Select * From __Namespace")
    For Each Namespace in Namespaces
             Set Providers = WbemServices.ExecQuery("Select * From __Provider")
            For Each Provider in Providers
                   key = "HKEY_CLASSES_ROOT\CLSID\" & Provider.CLSID
                   DLL=""
                   if Provider.CLSID<>"" Then
                          On Error Resume Next
                          DLL=shell.RegRead( key & "\InProcServer32\" )
                          If DLL="" Then DLL=shell.RegRead( key &
"\LocalServer32\" )
                          On Error Goto 0
                   End If
                   WScript.Echo Provider.Name & ", " & DLL
            Next
            GetNamespaces path & "\" & Namespace.Name, Level+1
    Next
End Sub
```

Your computer might have many providers defined, often in multiple namespaces. To identify each unique provider, the output from the previous script can be piped to the **SORT** command and directed to a text file for easier reading. This can be done with the following command:

```
Providers | Sort > Temp.txt
```

The following script provides a count of all the classes in each namespace:

```
'ClassCounts.vbs
Set loc = CreateObject("WbemScripting.SWbemLocator")
GetNamespaces "root", 0, "root"

Sub GetNamespaces(Path, Level, NamespaceName)
    Set WbemServices = loc.ConnectServer( , path )

    Set classes=WbemServices.SubClassesOf
    WScript.Echo Space(5*Level) & NamespaceName & ": " & classes.count

    Set Namespaces = WbemServices.ExecQuery("Select * From __Namespace")
    For Each Namespace in Namespaces
        GetNamespaces Path & "\" & Namespace.Name, Level+1, Namespace.Name
    Next
End Sub
```

Installation Issues

If all your WMI-based applications or tools do not work at all, it is possible that WMI is not properly installed. If the installation of the application or tool gives errors messages indicating that it cannot compile several MOF files, there is also a strong possibility that WMI did not get installed.

If you think that WMI might not be properly installed on the computer, check for the existence of the \WINNT\SYSTEM32\WBEM directory. If it is not there, WMI did not get installed during the installation of the operating system or management application (such as SMS). You can manually install the WMI core (as opposed to the WMI SDK). WMI is available for download from the Microsoft Web site at http://www.microsoft.com. Even if the WMI installation continues to fails, it might display error messages that help you correct the underlying problem.

Connectivity Issues

If a WMI-based application or tool does not work at all, it is possible that you cannot connect to WMI. This is especially true if you have verified that WMI is installed.

You should ensure that the WMI service is started. Also, ensure that DCOM and networking are enabled on the computer and working properly.

Use WMI Control (or Wbemperm.exe on earlier versions of WMI) to confirm that you have security permissions to access WMI and the namespace that you require.

Resource Consumption Issues

If your WMI-based management application or script is running successfully but very slowly, consider whether WMI is using an excessive amount of computer resources (such as memory or CPU time). In the worst cases, WMI resource consumption issues can cause the computer to respond slowly.

The **Processes** tab of Windows Task Manager (TaskMgr.exe) is an effective tool to confirm whether WMI is consuming excessive resources. In the **Image Name** column, review WinMgmt.exe to see if it is using a large percentage of **CPU** or a large amount of **CPU Time**. The **Mem Usage** column is also useful, but you should add the **VM Size** column to add virtual memory consumption to the display (by clicking **Select Columns** on the **View** menu). WMI usually uses 8 to 10 MB of memory or virtual memory.

Note

On computers running Windows XP or an operating system in the Windows Server 2003 family, the WMI service is run in a Svchost process.

The most common cause of WMI resource expenditure issues are WMI providers not running as designed. This is especially true for non-standard providers. Check to see if there are any known problems with the non-standard providers that are installed on your computer. A later version of the provider might be available.

Another common cause of WMI resource consumption issues is queries run against the CIM Repository. If the queries request a large amount of data or are poorly formed, WMI can use considerable resources to return the results. Queries built into your management application should be well tested so that they do not cause problems. However, any custom queries that you or other administrators have created might cause this problem. Also, resource consumption will be high if your queries are not run asynchronously.

Providers do not always run in the context of WinMgmt.exe. This is especially true with Windows XP and operating systems from the Windows Server 2003 family. If you suspect that a process consuming excessive resources might be running a WMI provider, you can use Tlist.exe (with the process identifier as the parameter) to list the DLLs that are loaded in that process. If you suspect a specific provider, you can use **Tlist -m providerDLLname** to determine which process it is running in. **Tlist** is part of the Windows 2000 Support Tools, which is available on the Windows 2000 product CD. **Providers.vbs**, shown in the "Verifying the State of the CIM Repository" section earlier in this appendix, lists the DLL for each provider.

Programming Issues

If an error is displayed while you are using an SMS script or tool, the error might include WMI-related details, such as an error code or number. Table B.5 lists the most common WMI errors. The description for each error might give you a better understanding of the problem so that you can correct the underlying cause. For less common WMI error messages, see the WMI SDK.

Table B.5 WMI Error Messages

Error code	Error number	Description
WBEM_E_FAILED	0x80041001	The call failed.
WBEM_E_NOT_FOUND	0x80041002	The object could not be found.
WBEM_E_ACCESS_DENIED	0x80041003	The current user does not have permission to perform the action.
WBEM_E_PROVIDER_FAILURE	0x80041004	The provider has failed at some time other than during initialization.
WBEM_E_TYPE_MISMATCH	0x80041005	A type mismatch occurred.
WBEM_E_OUT_OF_MEMORY	0x80041006	There was not enough memory for the operation.
WBEM_E_INVALID_CONTEXT	0x80041007	The **IWbemContext** object is not valid.
WBEM_E_INVALID_PARAMETER	0x80041008	One of the parameters to the call is not correct.
WBEM_E_NOT_AVAILABLE	0x80041009	The resource, typically a remote server, is not currently available.
WBEM_E_CRITICAL_ERROR	0x8004100A	An internal, critical, and unexpected error occurred. Report this error to Microsoft Product Support Services.
WBEM_E_INVALID_STREAM	0x8004100B	One or more network packets were corrupted during a remote session.
WBEM_E_NOT_SUPPORTED	0x8004100C	The feature or operation is not supported.
WBEM_E_INVALID_SUPERCLASS	0x8004100D	The superclass specified is not valid.
WBEM_E_INVALID_NAMESPACE	0x8004100E	The namespace specified could not be found.
WBEM_E_INVALID_OBJECT	0x8004100F	The specified instance is not valid.
WBEM_E_INVALID_CLASS	0x80041010	The specified class is not valid.
WBEM_E_PROVIDER_NOT_FOUND	0x80041011	A provider referenced in the schema does not have a corresponding registration.
WBEM_E_INITIALIZATION_FAILURE	0x80041014	A component, such as a provider, failed to initialize for internal reasons.
WBEM_E_TRANSPORT_FAILURE	0x80041015	A networking error that prevents normal operation has occurred.
WBEM_E_INVALID_OPERATION	0x80041016	The requested operation is not valid. This error usually applies to invalid attempts to delete classes or properties.
WBEM_E_INVALID_QUERY	0x80041017	The query was not syntactically valid.
WBEM_E_INVALID_QUERY_TYPE	0x80041018	The requested query language is not supported.

Learning More About WMI

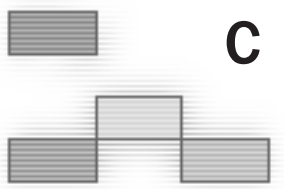
WMI is a sophisticated system. This appendix gives you enough information to manage WMI for SMS. Supplemented with the chapters listed in the "How SMS Uses WMI" section earlier in this appendix, you should be able to extend WMI and SMS to better serve your goals.

The following resources can give you a more extensive understanding of WMI:

- The WMI SDK, which is available at the Microsoft Web site at http://msdn.microsoft.com.
- The WMI Tutorial, which is available at the Microsoft Web site at http://www.microsoft.com/downloads/release.asp?ReleaseID=12570.
- An MSDN® News article, which is available at the Microsoft Web site at http://msdn.microsoft.com/library/. This is an administrator-oriented, general-interest introduction.
- An MSDN Magazine article, which is available at the Microsoft Web site at http://msdn.microsoft.com/msdnmag/issues/0400/wmi/wmi.asp. This is a programmer-oriented, general-interest introduction.
- The WBEM standard, which is available at http://www.dmtf.org/wbem/index.html.
- Third-party books about WMI.
- WMI newsgroups on msnews.microsoft.com. Use Outlook Express or a similar newsreader to access newsgroups.
- A third-party WMI Web site, which is available at http://desktopengineer.com/search.php?mode=search&type=stories&topic=0080WMI.

APPENDIX C

Scripting SMS Operations



Like any system, using Microsoft® Systems Management Server (SMS) 2003 extensively often requires performing numerous tasks many times. Most SMS tasks are usually performed by using the SMS Administrator console. However, you can write simple scripts to perform almost any SMS task. These scripts can then be repeated frequently or extended to work with many sites at once. Or, the scripts can run from features such as Scheduled Tasks in Microsoft Windows® 2000. This removes the need for an SMS administrator to be available when the task is required. Such scripts can save a lot of manual effort and can even make some things possible that would not otherwise be possible.

Using your imagination, a reasonable understanding of SMS, and the scripting details presented in this appendix, you can manage SMS more efficiently and create solutions to better serve your organization's needs. This appendix includes many practical examples, but these examples are only a small fraction of the possible solutions.

You do not have to be a programmer to write scripts. Programming skills can help, but simple scripts can be fairly intuitive for anyone. Using sample scripts (provided in this appendix and elsewhere), you can often accomplish tasks with only minor changes to the sample scripts. As you become more comfortable with scripting, your scripts can become more sophisticated.

The following are possible situations in which scripting SMS operations is useful:

- You have many sites in your SMS hierarchy and it is too tedious and error-prone to manually make changes to all the sites. It is also time-consuming to verify the settings on all sites.
- You are frequently creating, modifying, or removing SMS objects such as collections, advertisements, packages, queries, site systems, or reports.
- You have special needs when using SMS that are awkward or impossible for an SMS administrator to do with the SMS Administrator console or the available SMS tools.
- You have a task that must be completed for some of your SMS administrators, but it requires privileges that you do not want to give to those administrators — and it would be inconvenient for administrators with privileges to do it for them. In this case, you can set up a scheduled task to routinely do the task while running with a privileged account.

Scripts also have the benefit of being compact and are therefore easy to transfer and edit. They can be used with scheduling services, alert systems, and similar systems that can invoke scripts. You can even use SMS to run the scripts that automate your SMS operations. When you have fully debugged your scripts, there is no risk of error. An administrator creating objects might type or select values incorrectly, but a fully debugged script does it correctly every time.

In This Appendix

- Understanding Scripting
- Getting SMS Objects
- Working with SMS objects
- Working with SMS Site Settings
- Scripting Console Operations
- Scripting Client Operations
- Debugging Scripts
- Using Scripts on Web Pages
- Understanding Support Implications of Scripted Solutions
- Learning More

For information about creating scripts that can be used with SMS software distribution, which is not discussed in this appendix, see Chapter 5, "Distributing Software," and Chapter 7, "Creating Software Installation Packages with SMS Installer." For information about the scripting of SMS site setup, which is also not discussed in this appendix, see Chapter 15, "Deploying and Configuring SMS Sites," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Understanding Scripting

At the most fundamental level, SMS scripting is just a matter of creating a text file that includes a sequence of commands. When the script is ready, the file is executed in a scripting environment and (if the script is written correctly) the operation is performed.

A crucial first step for an SMS script is to connect to Windows Management Instrumentation (WMI). For information about WMI, see Appendix B, "Windows Management Instrumentation." SMS tools are almost always written to use the SMS Provider, which is a WMI provider and is the only way to ensure that SMS object security is enforced. Using the SMS Provider is also the only supported means to manage SMS. The SMS Administrator console uses the SMS Provider.

SMS scripting can be done in a variety of environments and languages. Any scripting environment that you can use to script WMI operations, you can use to script SMS operations. Many administrators find Windows Script Host (WSH) to be a readily available, easy-to-use, and powerful scripting environment. WSH is included with Microsoft Windows XP, all operating systems in the Microsoft Windows Server™ 2003 family, and Windows 2000 and is available for Microsoft Windows NT® 4.0, Windows 95, Windows 98, and Windows Millennium Edition.

On computers with WSH, you can use a script's full file name (and path, if necessary) as a parameter to the Cscript or Wscript commands, which are respectively console-oriented and window-oriented implementations of WSH. On computers running Windows XP, all operating systems in the Windows Server 2003 family, or Windows 2000, you do not have to use the Cscript or Wscript commands. To invoke your script, double-click the Cscript or Wscript command and enter the file name of your script (even without the file name extension) at the command prompt.

Note

Scripts are often written to display output at the command line. However, by default, WSH displays its output by using message boxes when you invoke scripts by double-clicking them or by entering the script file name at the command prompt. To change the default to command-line oriented output in those circumstances, use the command **Cscript //H:Cscript**.

WSH can support a variety of languages, including Microsoft Visual Basic® Scripting Edition (VBScript) and Microsoft JScript. VBScript scripts have the .vbs file name extension. VBScript is like any other scripting language, except that it also allows interfacing to Common Object Model (COM) objects, which are available for a large number of systems. Using COM objects, your scripts can work not only with WMI (and systems that it supports, such as SMS) but also with other systems that have scripting objects, such as Active Directory® directory services, Microsoft Exchange, and Microsoft SQL Server™.

Other environments that support the WSH languages are Dynamic HTML (DHTML) Web pages, Active Server Pages (ASP) Web pages, Visual Basic for Applications (VBA), as found in Microsoft Excel, and Visual Basic. With just a few changes, your SMS scripts can be adapted to work in each of these environments.

WSH scripting usually involves working with objects. This does not require an extensive familiarity with object-oriented programming. However, a familiarity with some object-oriented programming concepts, such as those listed in Table C.1, can help.

Table C.1 Key Object-Oriented Programming Concepts

Concept	Description
Object	A data element (like a variable), which has properties and methods.
Property	A value that applies to an object. For example, collections have name and comment properties (among others). For each collection object, you can see or set its name and comment.
Method	A function that does something relevant to an object. For example, for a process object, a method might be **create**, which creates processes. Using the **create** method, you could create a process that runs Notepad.exe. Many operations can be performed on objects by using standard scripting and WMI techniques. However, for complex operations, methods are often available.
Key property	A property that can be used to search for a specific object. For example, the **SMS_Collection** has a key property called **CollectionID**. If you have a collection identifier value, you can use it with the **CollectionID** key property to quickly find a specific collection, which is returned to your script as an object.

Your scripts will frequently work with WMI objects. In particular, the WMI objects will usually be SMS objects. WSH also has some objects of its own, particularly Wscript (with methods like **Echo** and **Quit**). **Wscript.Echo** is commonly used to display the value of properties. **Wscript.Quit** can be used to end the execution of a script; otherwise, scripts continue until the last line is executed. Wscript also has some classes that can be used to create objects. For example, your script can create an object from **Wscript.Network** that is used to collect details about the networking environment in which the script is running. This is shown in Sample C.9.

Writing Scripts

To write scripts, you have to be familiar with the syntax of the language in which you are planning to write the scripts. This is best accomplished by reading a book on the subject. However, you might be able to learn scripting by modifying samples to serve your purpose. Because most programming languages have an English-like syntax, you might be able to discern the meaning of the lines by just reading them if you are comfortable with English. For more information, see the "Learning More" section later in this appendix.

Important

As with other SMS tasks, you should test your scripts in a test environment before deploying them in your production environment. This prevents unexpected results from adversely affecting end users.

One approach to writing scripts is to use the following procedure.

A process for writing scripts

1. Think about what your script must accomplish. For example, it might need to connect to the SMS namespace (in WMI), find a collection that is based on a collection identifier, and change a property on that collection. Make a list or diagram of the flow of the script.
2. Execute the steps that are required to complete the operation by using the SMS Administrator console or a similar tool. Using Wbemtest.exe or CIM Studio from the WMI software development kit (SDK), view the WMI instances that were created by the operation. Watch for checks that the SMS Administrator console does to ensure that the operation is done successfully. For example, the SMS Administrator console does not allow you to create two collections with the same name. Ensure that the plan that was created in step 1 includes setting and checking all the values that are set and checked when the operation is performed manually.
3. Try to find a script that is similar to the script that you want to write. Relevant samples can be found in this appendix, the *Microsoft Systems Management Server 2003 Software Development Kit*, the WMI SDK, and many public Web sites. Even if the examples are not in the scripting language that you are using, they can still provide useful hints as to what you need to do. Copy the most relevant scripting sample that you can find. The script might have little in common with your intended script, but any common elements can help. For example, the script might connect to WMI or even to the SMS namespace. If you can find several scripts that have common elements with your intended script, copy the relevant parts to the new script.
4. Using Notepad or a similar editing program, remove lines of the copied script that are not similar to what you need. You might change them into comment lines if you suspect that you might need them in the future.
5. Starting with the first task that the script must accomplish, add or change lines to fit what you outlined in steps 1 and 2. You might start with a script that connects to the SMS namespace and then add lines to find the collection that you need. Do not add lines to modify the collection yet. This helps keep the script simple.
6. Run the script as it currently exists and correct any reported errors. Read relevant documentation and look at samples to find ideas for what might be wrong. If you still cannot correct the errors, simplify a copy of the script as much as possible while still reproducing the problem. This ensures that you are focusing on the lines that are causing the problem.
7. If possible, add a line to the script to display the results of the functionality that you added in step 4. For example, you might add a line to display the error status or collection name after you attempted to get the collection. This line is probably just temporary; you might not want it in your final script, but it can help to verify that your script is working properly so far.
8. Repeat steps 5, 6, and 7 until the script does everything that you outlined in steps 1 and 2.

9. Thoroughly test the script in your test environment. Verify that the script completely reproduces the changes that you observed in step 2. Test the script with a range of reasonable parameters to ensure that it behaves as expected. Test the script with invalid parameters and any other variations in sequence or environment that might be relevant to ensure that it rejects invalid parameters in a friendly manner and cannot do anything adverse. Have co-workers use the script to see if it works properly. Adjust the script until it safely does exactly what you require.
10. Test your script in a production environment. Start with a small, less significant site if you can and scale up your testing as you become confident that it is working exactly as you intend.
11. Put the script into production. This can include ensuring that all relevant details are documented for future reference and that all relevant co-workers and managers are aware of the script that you have implemented.

Caution

Your scripts manipulate SMS objects rapidly and directly, probably without human supervision. Under these circumstances, your scripts might do things that an SMS administrator would not do. For example, the SMS Administrator console checks for reasonable values (for example, that a collection has a unique name), but your script does not make such a check unless you add code to the script to do so.

Creating and Running a Simple Script

To ensure that you can create and run a simple SMS script, this section steps you through the process of creating and running a script that changes the name of a collection.

To create and run a simple SMS script

1. Outline what you want to accomplish with the script. In this case, the goal is to connect to SMS, find the "All Windows NT Systems" collection, and change its name to "All Windows NT Systems (any version, any type)".
2. Open Notepad, add the sample script shown in Sample C.1, and then save the file as C:\First.vbs.

Sample C.1 First.vbs — a sample script

```
Set loc = CreateObject("WbemScripting.SWbemLocator")
Set WbemServices = loc.ConnectServer( , "root\SMS\site_B2K")

Set Object = WbemServices.Get("SMS_Collection.CollectionID='SMS000CS'")
Wscript.Echo Object.CollectionID, Object.Name

Object.Name="All Windows NT Systems (any version, any type)"
object.Put_
```

In this example, because you have a script that is close to what you require, you do not have to remove any of the script's contents.

3. If you are working on the site server, the only thing you have to change is the site code. Change B2K to the site code for your test site server (and change it to the value for your production site server when the script is ready for production). If you are working on a computer other than a primary site server, change the second line to include the primary site server name and possibly a user name and password, such as shown in the following example.

```
Set WbemServices = loc.ConnectServer( "servername", "root\SMS\site_B2K"
"username", "password" )
```

4. Save the file again.
5. In the SMS Administrator console, look for the All Windows NT Systems collection, and then verify that you have a collection with this exact name. (Testing should always include verifying the state of the object before the script is run.)
6. Click the **Start** button, point to **Programs**, point to **Accessories**, and then click **Command Prompt**.
7. At the command prompt, type **Cscript C:\First.vbs**.
8. If there are any errors, correct them, save the file, and go back to step 6.
9. In the SMS Administrator console, click **Refresh**.
10. In the SMS Administrator console, look for the All Windows NT Systems collection. The name should now include "(any version, any type)". If it does not, correct any errors, save the file, and go back to step 6. Errors at this level are less obvious than the errors in step 10. They are either errors in the logic of the script (as opposed to its syntax) or problems outside of the script that prevent SMS from working properly.
11. If you are working on a computer running Windows 2000, Windows XP, or any operating system in the Windows Server 2003 family, you can try running the script by double-clicking it in Windows Explorer or by just entering the file name at the command prompt.

    This second execution of the script will not change the collection name (unless you write another script to change the collection name back to normal and run that script first). It will, however, allow you to see that scripts can be run from places other than the command prompt.

Note

If the site that you are testing in is a child site, the collection name in this example will be overwritten the next time the parent site sends an update for the collection.

Developing Scripts

The following are some advanced techniques that you can use to develop scripts:

- Browse current instances (using CIM Studio, for example) to see what the data should look like, what methods are available, and what class associations might be important.
- If the script is going to be complex, and especially if you do not have a close sample to start with, start the script development in Visual Basic to work out the details and the bugs. You can translate the script to VBScript later, if desired. In Visual Basic, you do not have to switch between your editor, run-time environment, and debugger, because Visual Basic serves all of these roles.,
- Use CIM Studio to see the details of the objects that you create or manipulate.
- If objects that you create or change in a script cannot be seen in the SMS Administrator console, but you can find them in WMI by using a tool like CIM Studio, your script might be missing details that the SMS Administrator console requires. For example, collections require a pointer to the collections. For more information, see the "Collection Creation Example" section later in this appendix.
- Review relevant documentation, such as the Microsoft Systems Management Server 2003 Software Development Kit, for detailed information about subtle concepts related to SMS scripting. For a list of SMS resources, see the "Learning More" section later in this appendix.

Scripting in Visual Basic

If you choose to start writing your script as a Visual Basic program, use the following procedure.

To create a Visual Basic program for SMS operations

1. Start Visual Basic, and then create a new project.
2. On the **Project** menu, click **References**, and then click **Microsoft WMI Scripting Vx.x Library**.
3. In the "(General)", "(Declarations)" section of your program, add `Private WbemServices as SWbemServices.`
4. In a new subroutine (or function), add `Dim loc as New SWbemLocator.`
5. Add `Set WbemServices = loc.ConnectServer( , "root\SMS\site_<site code>")` (add the server name as the first parameter and the user name and password as the third and fourth parameters (if necessary) if the script will run from a computer other than the site server).
6. Wherever you need variables for the WMI objects, include `Dim <variable name> as SWbemObject` (or `SwbemObjectSet`).

The Visual Basic program is now ready to work with WMI, and thus with SMS objects. Examples from this appendix can be used in the program (except that the lines to create a locator and connect to the WMI server do not have to be included).

If you are going to take advantage of Wscript classes (such as (like **Wscript.Network** or **Wscript.Shell**) in your scripts, add a reference for Windows Script Host Object Model and use it like the locator (for example, `Dim X as New IWshNetwork_Class`).

If you created the program in Visual Basic and want to translate it to VBScript (so that it can be used wherever scripts are used), use the following procedure.

To translate scripts developed as Visual Basic programs to VBScript

1. Copy the subroutines from Visual Basic.
2. Change the line from step 4 in the previous procedure to `Dim loc`.
3. Add `Set loc=CreateObject("WbemScripting.SWbemLocator").`
4. The line that was added in step 5 of the previous procedure stays the same.
5. Change lines that were added in step 6 to `Dim <name>` (and do not specify a data type).
6. Where appropriate, translate any logic that outputs values (by adding them to list boxes, for example) to use **Wscript.Echo** instead.
7. Where appropriate, add lines to create any objects that your Visual Basic program references, such as:

```
Dim x As WshNetwork
Set x = New WshNetwork
```

Must be changed to:

```
Set x=CreateOject("Wscript.Network"
```

Connecting to WMI

The first thing that your scripts will have to do is connect to WMI, which is shown in Sample C.2.

Sample C.2 Connect.vbs — the part of a script that connects to WMI

```
Dim loc
Set loc = CreateObject( "WbemScripting.SWbemLocator" )
Dim WbemServices
Set WbemServices = loc.ConnectServer( ,"root\SMS\site_B2K" )
```

Only the fourth line ever changes. The B2K value is the site code of a primary site and it varies depending in which site you are running your script.

When the fourth line includes only the second parameter (“root\SMS\site_<sitecode>”), the script is intended to run on the site server. WMI does not accept any other parameters on that line when the script is run on the site server.

If you want to run your script from another computer (an SMS Administrator console, for example), your script still has to connect to the site server. However, to do that, the fourth line has to include a server name as its first parameter, as follows:

```
Set WbemServices = loc.ConnectServer( "servername", "root\SMS\site_B2K")
```

When connecting remotely, you can also specify a user name and password, as follows:

```
Set WbemServices = loc.ConnectServer( "servername, "root\SMS\site_B2K",
"username", "password")
```

If you need to specify a domain for the user name, the line changes as follows:

```
Set WbemServices = loc.ConnectServer( "servername, "root\SMS\site_B2K",
"domain\username", "password")
```

Because the lines in Sample C.2 are used in all scripts in this appendix (except where otherwise noted) and they do not vary dramatically, some of the scripts in this appendix do not include these lines.

WMI supports multiple models for connecting to WMI from scripts. These models include the moniker and moniker session models, which allow you to connect to WMI by using fewer lines of code than the model discussed in this section. However, because these models are less flexible, most examples in this appendix do not use them. For more information about those models, see the WMI SDK.

For more advanced connection code that eliminates the need for referencing a site code, see the *Microsoft Systems Management Server 2003 Software Development Kit*.

Getting SMS Objects

Now that you are comfortable developing and using scripts, you can concentrate on the mechanics of SMS scripting. The best place to start is by retrieving SMS data. Such scripts should not change any SMS configuration details, and therefore you are not at risk of causing adverse effects.

Getting SMS objects is often the first thing that any script you write has to do. Until you get objects, you cannot change or delete them. However, getting SMS objects in a script can also be the easiest way to get information from SMS. When using a script, you do not have to open Microsoft Management Console (MMC) and the SMS Administrator console and then navigate through the console tree. If you keep a command prompt window open (as many administrators do), all you have to do is enter the name of the script and possibly some relevant parameters.

Note

WMI scripting can be done with several different WMI scripting models. All examples in this appendix use the WMI session model (unless otherwise noted), which is the most flexible model. For more information about WMI scripting models, see the WMI SDK. Also, not all script examples include the lines to create the locator and connect to the server. You can copy these lines from scripts that do have them.

The most common and flexible method to retrieve SMS data in a script is to execute a query, as shown with the following line:

```
Set Machines = WbemServices.ExecQuery( "Select * From SMS_R_System" )
```

The query syntax is the same as what is used in the SMS Administrator console, which uses the same WMI Query Language (WQL) syntax.

If you need to retrieve a particular instance (not a set of instances), and the value for looking up the instance is contained in a key property, it is faster and more efficient to get the instance (instead of querying for it), as shown in the following line:

```
Set Machine=WbemServices.Get( "SMS_R_System.ResourceID=5" )
```

For information about the available SMS classes and their keys, see the Class Schema Reference in the *Microsoft Systems Management Server 2003 Software Development Kit*. The best way to understand the SMS classes is to browse them, as described in Appendix B, "Windows Management Instrumentation."

If you cannot use the WMI **Get** method (because you do not have the value of a key property to find the instance you need), you have to do a query and enumerate the set of returned results and use the details.

Sample C.3 shows querying for SMS objects from a Visual Basic program. Sample C.4 is the equivalent WSH script.

Sample C.3 Query1.bas — a Visual Basic program that uses a query to get SMS objects

```
Private WbemServices As SWbemServices
Private Sub Form_Load()
    Dim loc As New SWbemLocator
    Set WbemServices = loc.ConnectServer(, "root\SMS\site_B2K")
       Dim Machine As SWbemObject
       Dim Machines As SWbemObjectSet
       Set Machines = WbemServices.ExecQuery("Select * From SMS_R_System")
       For Each Machine In Machines
           MsgBox "Got system " + Machine.Name +": "+ Machine.IPAddresses(0)
       Next
End Sub
```

Sample C.4 Query1.vbs — a WSH equivalent to Sample C.3

```
Dim loc
Set loc = CreateObject("WbemScripting.SWbemLocator")
Dim WbemServices
Set WbemServices = loc.ConnectServer( ,"root\SMS\site_B2K")

Dim Machine
Dim Machines
Set Machines = WbemServices.ExecQuery("Select * From SMS_R_System")
For Each Machine In Machines
    MsgBox "Got system " + Machine.Name
Next
```

Note

Remember to change the B2K site code in the previous two examples to the site code for your site. Also, the examples are intended to run on the site server itself. If you want to run them remotely, you have to add the server and possibly the user name and password details on the ConnectServer line, as described in the "Connecting to WMI" section earlier in this appendix.

When returning results in your script, some properties might require special treatment, such as date or array properties. You might have to include code to enumerate the array elements or to parse the date values. The samples in this appendix include examples of how to work with properties.

If you are returning more than one class in your query, you must be more specific when referring to the properties. For example, when using the results from Sample C.5, you must refer to the returned properties as **Machine.CS.Name** and **Machine.NW.MacAddress** (as opposed to **Machine.Name** and **Machine.MacAddress**).

Sample C.5 Query2.vbs — part of a script to query two classes at once

```
Set Machines = gService.ExecQuery("Select CS.Name, NW.MACAddress From
SMS_G_System_COMPUTER_SYSTEM CS, SMS_G_System_NETWORK_ADAPTER NW WHERE
CS.Name='"+ computername +"' AND NW.ResourceID=CS.ResourceID")
```

Reporting Script

Using a script to get SMS objects can be a powerful form of SMS reporting. Scripts have complete flexibility when manipulating the data that is returned by the queries. You can supplement the data with other sources, do extensive calculations on the data, find complex relationships in the data, and do many other operations on the data that you cannot do using even powerful reporting tools. Also, scripts can retrieve lazy properties, which query and report tools cannot do. For more information, see the "Retrieving Lazy Properties" section later in this appendix.

Formatting the report with many fonts, graphics, and other attractive elements can be tedious in a script, but you can format the script output in such a way that it can be used in a reporting tool where such elements are easily added. For example, you could output the script's results into a comma-delimited file and then import that file into Microsoft Access for the final formatting.

Sample C.6 returns the version of all sites in your SMS hierarchy. Note that the use of recursion produces output that reflects your SMS hierarchy. The multiple IF statements allow you to translate SMS build numbers into meaningful results.

Sample C.6 SiteVersion.vbs — a hierarchical report of sites and their versions

```
Sub Level( parent, depth, extended_Description )
    Dim Site
    Dim Sites
    Set Sites = WbemServices.ExecQuery("Select * From SMS_Site Where
ReportingSiteCode='"+parent+"'")
    For Each Site In Sites
            zBuild=Site.BuildNumber
             If Site.BuildNumber<692  Then zBuild="SMS 1.0"
             If Site.BuildNumber=692  Then zBuild="SMS 1.1"
             If Site.BuildNumber=786  Then zBuild="SMS 1.2"
             If Site.BuildNumber=1239 Then zBuild="SMS 2.0 (no SP)"
             If Site.BuildNumber>1239 AND Site.BuildNumber<1380 Then zBuild="SMS
2.0 pre-SP1 interim build (" & Site.BuildNumber & ")"
             If Site.BuildNumber=1380 Then zBuild="SMS 2.0 SP1"
             If Site.BuildNumber>1380 AND Site.BuildNumber<1493 Then zBuild="SMS
2.0 pre-SP2 RC1 interim build (" & Site.BuildNumber & ")"
             If Site.BuildNumber=1493 Then zBuild="SMS 2.0 SP2 RC1"
             If Site.BuildNumber>1493 Then zBuild="SMS 2.0 post-SP2 RC1 (" &
Site.BuildNumber & ")"

             Descr=Site.SiteName + " ("+Site.SiteCode+") "
            Descr=SPACE( depth*3 ) + Descr
            Descr=Left( Descr, 35)
            Descr=Descr + SPACE(36-LEN(Descr))

            WScript.Echo Descr & zBuild
             extended_Description=extended_Description + Chr(13) + Descr & zBuild

            Level Site.SiteCode, depth+1, extended_Description
    Next

End Sub

    Dim loc
    Set loc = CreateObject("WbemScripting.SWbemLocator")
    Dim WbemServices
    Set WbemServices = loc.ConnectServer( , "root\SMS\site_CEN")

    'start with the central site
    Dim xDescr
    xDescr=""
    Level "", 0, xDescr
```

The results of this script will be something like the following:

```
Lab Central (CEN)                  SMS 2.0 SP2 RC1
   Microsoft (B2K)                 SMS 2.0 pre-SP2 RC1 interim build (1459)
      Beige, Secondary (BSE)       SMS 2.0 pre-SP2 RC1 interim build (1472)
   Litware, Inc. (LIT)             SMS 2.0 pre-SP2 RC1 interim build (1489)
   Lab Project 1 (PR1)             SMS 2.0 SP2 RC1
      RED2 - Child of RED (RD2)    SMS 2.0 SP2 RC1
   Lab Project 2 (PR2)             SMS 2.0 SP2 RC1
   Lab SMS 1.2 (S12)               SMS 1.2
   Lab Windows 2000 Experiment     SMS 2.0 SP2 RC1
```

Displaying Distribution Point Status

Sample C.7 shows another example of a script that produces a report. This report can be done with a reporting or query tool, but the script easily parses the returned results into a more readable format.

Sample C.7 LogonPointsDown.vbs — a report of any logon points that are down

```
winmgmt1 = "winmgmts:{impersonationLevel=impersonate}!//" & objArgs(0) &
"\root\sms\site_" & sitecode
Set objWMI = GetObject( winmgmt1 )

WScript.Echo "Site Code, Site System, Down Since"
strQuery = "select SiteCode, SiteSystem, DownSince from SMS_SiteSystemSummarizer
where Role LIKE '%distribution%' AND NOT  Status = 0 order by SiteCode,
SiteSystem"
Set objEnumerator = objWMI.ExecQuery(strQuery)
for each instance in objEnumerator
       WScript.Echo instance.SiteCode & "," & Left(Mid(instance.SiteSystem, 13,
20), InStr(1, Mid(instance.SiteSystem,  13, 20),"\") -1) & "," &
instance.DownSince
       WScript.Echo instance.SiteCode & "," & Left(Mid(instance.SiteSystem, 13,
20), InStr(1,  Mid(instance.SiteSystem, 13, 20),"\") -1) & "," &
instance.DownSince
Next
```

Retrieving Lazy Properties

Some SMS object properties are relatively inefficient to retrieve. If these properties were retrieved for many instances in a class (as might be done in a query), the response would be considerably delayed. Such properties are considered lazy properties and are not retrieved during query operations. If these properties are retrieved during a query, they have null or zero values, which might not be the actual value of the property for every instance. If you want to get the correct value for lazy properties, you must get each instance individually. For information about which properties are lazy, see the SMS class definitions in the *Microsoft Systems Management Server 2003 Software Development Kit* .

Sample C.8 shows how to retrieve lazy properties. The assigned schedule properties for advertisements are lazy. Therefore, when querying advertisements, each advertisement instance must be retrieved separately and the assigned schedule properties must be parsed separately.

Sample C.8 Lazy.vbs — retrieving properties for advertisements, including lazy properties

```
Set Advertisements = WbemServices.ExecQuery("Select * From SMS_Advertisement")
For Each Ad In Advertisements
    WScript.Echo "ActionInProgress = " & Ad.ActionInProgress
    WScript.Echo "AdvertFlags = " & Ad.AdvertFlags
    WScript.Echo "AdvertisementID = " & Ad.AdvertisementID
    WScript.Echo "AdvertisementName = " & Ad.AdvertisementName

    'for the 'lazy' properties, get the advertisements individually
    Set lazyproperties = WbemServices.Get("SMS_Advertisement.AdvertisementID='"
& Ad.advertisementid & "'")
    WScript.Echo "Assigned Schedule = {"
       for i=0 to ubound(lazyproperties.assignedschedule,1)
                  WScript.Echo " instance of " &
lazyproperties.Properties_("AssignedSchedule").Value(0).Path_.Class
      WScript.Echo
lazyproperties.Properties_("AssignedSchedule").Qualifiers_("CIMType")
                    WScript.Echo "   DayDuration: " &
lazyproperties.AssignedSchedule(i).DayDuration
                    WScript.Echo "   Hourspan: " &
lazyproperties.AssignedSchedule(i).HourSpan
                    WScript.Echo "   IsGMT: " &
lazyproperties.AssignedSchedule(i).IsGMT
                    WScript.Echo "   StartTime: " &
lazyproperties.AssignedSchedule(i).StartTime
            WScript.Echo "AssignedScheduleEnabled = " &
lazyproperties.AssignedScheduleEnabled
       WScript.Echo "AssignedScheduleIsGMT = " &
lazyproperties.AssignedScheduleIsGMT
   next
    WScript.Echo "}"

    WScript.Echo "CollectionID = " & Ad.CollectionID
    WScript.Echo "Comment = " & Ad.Comment
Next
```

Advanced Queries

Queries are usually the simplest form of scripting, and because they only return data (rather than changing it), they are also the safest form. However, in some cases queries can be advanced.

Large Queries

When managing large amounts of SMS data, some of the scripts in this appendix should be optimized for efficiency. This is especially true for scripts that query SMS, because queries almost always return multiple instances. The primary method for making scripted SMS operations more efficient is to run them asynchronously. Running the operations asynchronously allows the server to send data to your script in manageable portions, instead of using a large amount of memory on the server to collect all the results before sending them to your script in one large amount.

The primary WMI methods have asynchronous versions. For more information, see the WMI SDK. In the case of queries, you can use the **ExecQueryAsync** method, rather than **ExecQuery**.

Sample C.9 AsyncQuery.vbs — executing queries asynchronously

```
Set service = GetObject("winmgmts:root\SMS\site_MSO")
Set sink = WScript.CreateObject("WbemScripting.SWbemSink","SINK_")

service.ExecQueryAsync sink, "Select * FROM SMS_Collection"

'make a call or do work that prevents the script from ending before all the
events are received.
MsgBox "Waiting for instances."

Sub SINK_OnObjectReady(objObject, objAsyncContext)
    WScript.Echo "A collection has been returned: ", objObject.Name
End Sub

Sub SINK_OnCompleted(iHResult, objErrorObject, objAsyncContext)
    if iHResult=0 Then
           WScript.Echo "All collections returned."
    else
            WScript.Echo "There was a problem: ", Hex(iHResult)
    end if
End Sub
```

Limited Queries

Another method for making SMS scripted operations more efficient is to limit the returned results. This ensures that the operations do not use excessive resources if your script has a problem. Limiting scripted operations can also be useful for quick tests.

To limit scripted SMS operations, use SMS context qualifiers. For more information about SMS context qualifiers, see the *Microsoft Systems Management Server 2003 Software Development Kit*. You can specify an "InstanceCount" qualifier to force the operations to return a limited number of results, as shown in Sample C.10. You can also limit the operations to specific collections by using the "LimitToCollectionIDs" qualifier, so that only resources in specified collections are used.

Note

You should also set the "ApplicationName", "LocaleID", and "MachineName" qualifiers when using SMS context qualifiers. These qualifiers allow you to audit operations in your SMS sites. For security reasons, you might want to set these values even when you are not limiting the SMS operations.

Sample C.10 ControlQuery.vbs — limiting queries

```
Dim WbemContext
Set WbemContext=CreateObject("WbemScripting.SWbemNamedValueSet")
WbemContext.Add "ApplicationName", "ControlQuery.vbs"
WbemContext.Add "LocaleID", "ms\0x0409"
WbemContext.Add "MachineName", "Machine44"
WbemContext.Add "InstanceCount", 3

Set loc = CreateObject("WbemScripting.SWbemLocator")
Set WbemServices = loc.ConnectServer( , "root\SMS\site_B2K")
Set Collections = WbemServices.ExecQuery("Select * From SMS_Collection", , ,
WbemContext)
For Each Collection In Collections
    Wscript.Echo Collection.Name
Next
```

Status Queries

SMS status can be queried from scripts in the same manner that other SMS objects are queried. The Structured Query Language (SQL) statements for the predefined status message queries in the SMS Administrator console can be used as examples for queries that can be run from scripts to return SMS status messages.

To query site component or site detail status, you must query the **SMS_ComponentSummarizer** or **SMS_SiteDetailSummarizer** class, respectively. For these classes, you must specify both the site code and a tally interval. *Tally intervals* are values such as "0001128000080008", which returns the status since the site was installed. Other tally intervals can be used, for example, to check the status for the last week. For more information about tally interval values, see the *Microsoft Systems Management Server 2003 Software Development Kit*.

Working with SMS Objects

You can create, modify, or delete SMS objects. Depending on the purpose of your script, your script might do a combination of these operations. For example, it might get an object, create a new one that is similar, and then delete the original. Your script might also repeat these operations many times. Your scripts will primarily consist of this limited set of operations.

To create SMS objects

1. Get the class that represents the objects.
2. Launch an instance of the class.
3. Set additional details (properties) on the instance.
4. Save the instance.

Assigning a unique identifier to the new object is not a step in the process of creating an SMS object. SMS automatically assigns unique identifiers to objects when they are saved.

To modify an object

1. Get an instance of the object.
2. Adjust the details (properties) on the instance.
3. Save the instance.

To delete an object

1. Get an instance of the object.
2. Delete it.

The first step of each of these operations (getting an instance of an object) was discussed in the "Getting SMS Objects" section earlier in this appendix. This section discusses how to launch instances, set properties, and delete objects.

Later sections discuss specific issues about working with collections, advertisements, and packages, but the same principles can be applied to all SMS objects.

Note

SMS objects are considered to be:

- Collections.
- Advertisements.
- Packages (including programs and the distribution points that are used by each package).
- Security credentials.
- Queries.
- Reports.
- Status queries.
- Resources.
- Distribution point groups.

Site details (including site settings, client agents, status filter rules, and site systems) are not considered SMS objects. For information about scripting site details, see the "Working with SMS Site Settings" section later in this appendix.

Collections

Collections are fundamental to the use of SMS. They are important for software distribution and are often used to find computers when using SMS Remote Tools. Collections can also be used for limiting the scope of queries.

Scripting collection operations can ease your use of SMS in many circumstances, including:

- Creating collections automatically. For example, a script could create one collection per site or add resources that are listed in an input file to a collection.
- Adding resources to a collection that a query could not find. This is especially useful if it takes too long to manually add resources to the collection. For example, a script could add all the users who are in both of two particular Windows NT groups to a collection. Such a script would obtain the members of the Windows NT groups, find the members that are common to both groups, and then add those members to the collection.
- Setting security automatically. For example, a script could make collections available to everyone in the group, not just to the creators of the collections. The script could be automatically run once every few hours as a Windows task. When you set up the task, you can set it to use an account with sufficient privileges to change collection security. With this task in place, you do not have to give the privileges to the people who create collections (which you might not want to do because those privileges would also allow them to do other functions that you might not want them to do).
- Determining which collections a computer was assigned to and making another computer a member of those collections. A script that does this would be very helpful when the original computer is replaced.
- A Web page that could let users request software distributions to themselves. The script would add the users to an appropriate collection that has the package advertised to it.

Creating collections follows the same pattern as creating any other SMS object; get the collections class, launch an instance, set some properties, and then save it. Collections are exceptional, however, in that they also have two subobjects:

Collection-to-collection relationships These objects allow collections to have subcollections. Every collection is considered a subcollection to the master collection that the SMS Administrator console uses to display collections. The master collection's collection identifier is COLLROOT. This subobject class is **SMS_CollectToSubCollect**.

Collection membership rules Collection membership rules can either be direct assignment rules (specific computers, users, or user groups) or they can be queries. The queries are executed any time the collection is updated and any discovered computers, users, or user groups that meet the criteria of the queries are considered part of the collection. The subobject classes are **SMS_CollectionRuleDirect** and **SMS_CollectionRuleQuery**, respectively.

Collection Creation Example

Sample C.11 shows creating a collection, the collection-to-collection relationship, and a direct membership rule. It also shows how the collection-to-collection and collection membership rules are associated with the collection. The sample script takes one parameter, which is a collection name. You can use any valid collection name. The script adds the user that is running the script to the collection. This script can be used on a Web page to add the current user to a collection that has software advertised to it. Users can then initiate their own software distributions.

Sample C.11 CreateCollection1.vbs — creates a collection and adds the user running the script to the collection

```
'get a command-line argument - this makes the script flexible
if wscript.arguments.count<>1 then
    WScript.Echo "One argument please - the collection name"
    WScript.Quit
else
       newname = wscript.arguments(0)
end if

'connect to WMI and the SMS namespace, as usual
Set lLocator = CreateObject("WbemScripting.SWbemLocator")
Set gService = lLocator.ConnectServer( , "root\sms\site_B2K")

'check if the Collection name specified is unique
alreadyused = False
Set Collections = gService.ExecQuery("Select * From SMS_Collection")
For Each Collection In Collections
    If Collection.Name = newname Then alreadyused = True
Next
If alreadyused Then
   MsgBox "This collection name is already in use. Please enter a different
name."
Else

   'create the collection
   Dim newCollection
   Set newCollection = gService.Get("SMS_Collection").SpawnInstance_()
   newCollection.Name = newname
   newCollection.OwnedByThisSite = True
   newCollection.comment = "this is just a test"
   newCollection.Put_
   path=newCollection.Put_
   'and get the automatically assigned collection ID for our new collection
   Set Collection=gService.Get(path)
   newcollectionid= Collection.CollectionID
```

(continued)

Sample C.11 CreateCollection1.vbs — creates a collection and adds the user running the script to the collection *(continued)*

```
   'create the collection relationship
   'you could use the VerifyNoLoops method of the SMS_Collection class if you
   'want to ensure that you won't create a loop of collections
   Dim newCollectionRelation
   Set newCollectionRelation = gService.Get( "SMS_CollectToSubCollect"
).SpawnInstance_()
   newCollectionRelation.parentCollectionID = "COLLROOT"
   newCollectionRelation.subCollectionID = newcollectionid
   newCollectionRelation.Put_

   'determine the current user, as a resource ID
   Set oWshNetwork=CreateObject("Wscript.Network")
   username = oWshNetwork.UserName
   Set Users = gService.ExecQuery("Select * From SMS_R_User WHERE Name LIKE ""%"
+ username + "%""")
   For Each User In Users
      If User.username = username Then
         ResID = User.ResourceID
      End If
   Next
   'you might want to handle the contingency of a user that is new to
   'the domain and hasn't been added to the list of SMS users yet

   'create a direct collection rule for the user we just determined
   Set CollectionRule =
gService.Get("SMS_CollectionRuleDirect").SpawnInstance_()
   CollectionRule.ResourceClassName = "SMS_R_User"
   CollectionRule.RuleName = "ResourceID=" & ResID
   CollectionRule.ResourceID = ResID

   'add the rule to the collection
   Collection.AddMembershipRule CollectionRule
   If Err.Number = 0 Then
      Wscript.Echo "You were added to the " + Collection.Name + " collection!"
   End If

End If 'this is the end of the test for a unique collection
```

Adding a query rule to a collection is similar to adding a direct resource rule, as shown in Sample C.12. In Sample C.12, a different class is used (**SMS_CollectionRuleQuery**) and a query is added instead of a resource identifier. Also, if you do not want to wait for the collection evaluator to find members that qualify for the collection, you can refresh it from the script by using the **RequestRefresh** method. This is the equivalent of using the **Update Collection Membership** menu item in the SMS Administrator console.

Sample C.12 CreateCollection2.vbs — a partial script where the collection rule is a query rule rather than a direct rule

```
   sitecode="B2K"
   'create a direct collection rule for the user we just determined
   Set CollectionRule = gService.Get("SMS_CollectionRuleQuery").SpawnInstance_()
   CollectionRule.RuleName = "All clients that are assigned to site code " &
sitecode
   CollectionRule.QueryExpression = "select * from SMS_R_System where
SMSAssignedSites = '" & sitecode & "'"

   'add the rule to the collection
   Collection.AddMembershipRule CollectionRule
   If Err.Number = 0 Then
      WScript.Echo "The query was added to the " + Collection.Name + "
collection"
      Collection.RequestRefresh
   End If
```

Using Class-Specific Methods

Some SMS classes have methods that can be used for class-specific operations. For example, in Samples C.9 and C.10, the collection membership rule was added to a collection by using the collection-specific **AddMembershipRule** method. For more information about the SMS class methods, see the *Microsoft Systems Management Server 2003 Software Development Kit*. You can also see method details when browsing classes with the CIM Studio. Some documentation about these methods is also available by using the **Help for class** (**?**) button in CIM Studio while the class is selected.

Some class-specific methods are used on the class as a whole and are known as *class methods*. Others methods are used on specific instances of the class and are known as *instance methods*. For more information about these methods, see the *Microsoft Systems Management Server 2003 Software Development* Kit. The **AddMembershipRule** method is an example of an instance method and is used with a specific instance of the **SMS_Collection** class.

The **GetTotalNumResults** method, as used in Sample C.13, is an example of a class method. Sample C.13 shows using the **GetTotalNumResults** method to get the number of unique resources in a collection and its subcollections. This is similar to the **Show Count** menu command that is available for collections in the SMS Administrator console, except that menu command does not include subcollections.

Sample C.13 CollectionCount.vbs — part of a script to display the number of resources in a collection and its subcollections

```
Set CollectionClass=gService.Get("SMS_Collection")
CollectionClass.GetTotalNumResults collectionID, count
WScript.Echo count
```

Removing Rules from a Collection and Deleting Collections

Creating collections consists of creating the collection, collection-to-collection rules, and membership rules. However, the membership rules are embedded in the collection instance and the collection-to-collection rules are automatically deleted by SMS when the collection is deleted. Therefore, deleting a collection only involves one line of code.

```
Collection.Delete_
```

As is the case with the SMS Administrator console, deleting the collection does not result in the members of the collection being deleted.

To remove a membership rule from a collection, you use the **SMS_Collection DeleteMembershipRule** method. You use this method only if you want to remove a specific member from a collection but otherwise want leave the collection intact. Sample C.14 shows how to remove a membership rule from a collection. It also shows how to list membership rules for all collections.

Sample C.14 RemoveRule.vbs — part of a script to remove a rule from a collection

```
Set Collections = gService.ExecQuery("Select * From SMS_Collection")
For Each Collection In Collections
       'collection rules are a lazy property, so we must get them
   Set Collection=gService.Get("SMS_Collection='" & Collection.CollectionID &
"'" )
       Wscript.Echo Collection.Name
       If Not (IsNull(Collection.CollectionRules)) Then
                  Rules=Collection.CollectionRules
                  For Each Rule in Rules
                             Wscript.Echo "   ", Rule.Rulename, "  type:",
Rule.Path_.Class
                             If Rule.Rulename="All client machines that report
being assigned to site code " & sitecode Then Collection.DeleteMembershipRule
Rule
                  Next
       End If
Next
```

Deleting Resources

You delete resources (such as computers, users, or user groups) from SMS the same way that you delete SMS objects. You get a resource and then delete it, as shown in Sample C.15.

Sample C.15 DeleteResource.vbs — part of a script to delete a resource (a user)

```
Set User = gService.Get("SMS_R_User='" & resourceID & "'")
User.Delete_
```

You can delete all resources in a collection by using the **SMS_Collection** class **DeleteAllMembers** method, which is shown in Sample C.16. This is a powerful but dangerous method, so you must use it carefully. You can easily delete all the clients from your SMS database with it. Discovery methods can repopulate the information, but this can take some time and be a considerable workload on your systems and network.

Sample C.16 DeleteAllResources.vbs — part of a script to delete all resources (not just their membership) in a collection

```
Set collection = gService.Get("SMS_Collection='" & collectionID & "'")
message = "a strong warning message for " & collection.name
return=MsgBox( message, vbYesNo, "Delete_Users.vbs" )
if return=vbYes Then
       Collection.DeleteAllMembers
   WScript.Echo "All resources for that collection have been deleted."
    End If
```

When resources are deleted from SMS, SMS automatically removes all relevant records, such as computer hardware details, software inventory details, and history data. Therefore, your script does not have to clean up that data.

Advertisements

Scripting advertisement-related operations can be useful for many purposes, including:

- Automatically advertising new software as it becomes available (antivirus updates, for example). This should be done only with software that comes from a highly trusted source and that does not change dramatically over time. Usually, software should be tested thoroughly before advertising it to end users or production servers.
- Disabling an advertisement during working hours.
- Unlocking advertisements that originated at a parent site when a site is removed from its parent.

Scripting with advertisements is similar to scripting with collections. However, because advertisements do not have a hierarchy (with subadvertisements, for example), there is no need for advertisement-to-advertisement relationships. Similarly, advertisements often do not have subobjects like membership rules, though they might have assignments. Therefore, creating advertisements requires only launching an instance, setting properties (including program ID and collection ID), and then saving the instance. However, some of the advertisement properties are complex. These include expiration time and presentation time (**PresentTime**), if used. These values can also be difficult to calculate.

Creating Advertisements

Sample C.17 shows a script that creates a collection. You specify the package name, program name, collection name, and advertisement name. The package, program, and collection must already exist. For example, you can enter the following:

```
CreateCollection "Notepad" "RunIt" "All Systems" "Notepad Advertisement 1"
```

Note

Sample C.17 also includes some date manipulation code. It adds today's date and today's date plus six months into WMI-compatible variables. That code might be useful in any script that requires dates.

Sample C.17 CreateAdvertisement.vbs — a script to create an advertisement

```
if wscript.arguments.count<>4 then
       WScript.Echo "Four arguments please:"
       WScript.Echo "    the package name, program name, collection name and
advertisement name"
       WScript.Echo "    the parameters are space delimited (spaces between
them)"
       WScript.Echo "    package names do not include vendor, version, or
language"
       WScript.Echo "    the ad name is for the new advertisement - the rest
must exist already"
              WScript.Echo "    the names are case sensitive"
       WScript.quit
else
       PackageName    = wscript.arguments(0)
       ProgramName    = wscript.arguments(1)
       CollectionName = wscript.arguments(2)
       AdName         = wscript.arguments(3)
end if

Set loc = CreateObject("WbemScripting.SWbemLocator")
Set WbemServices = loc.ConnectServer( , "root\SMS\site_B2K")

PackageFound=false
ProgramFound=false
CollectionFound=false
AdvertisementFound=false

Set Packages = WbemServices.ExecQuery("Select * From SMS_Package Where Name='" &
PackageName & "'")
For Each Package In Packages
       WScript.Echo "Found the Package"
    PackageFound=true
    PackageID=Package.PackageID
    On Error Resume Next
    WbemServices.Get("SMS_Program.PackageID='" & PackageID &"',ProgramName='" &
ProgramName & "'")
    If Err=0 Then
          WScript.Echo "Found the Program"
       ProgramFound=true
    End If
```

(continued)

Sample C.17 CreateAdvertisement.vbs — a script to create an advertisement *(continued)*

```
       On Error Goto 0
Next

Set Collections = WbemServices.ExecQuery("Select * From SMS_Collection")
For Each Collection In Collections
       If Collection.Name=CollectionName Then
            WScript.Echo "Found the Collection"
            CollectionFound=true
                  CollectionID=Collection.CollectionID
       End If
Next

Set Advertisements = WbemServices.ExecQuery("Select * From SMS_Advertisement")
For Each Advertisement In Advertisements
       If Advertisement.AdvertisementName=AdName Then
                  WScript.Echo "Duplicate Advertisement name, but that's
allowed"
                  AdvertisementFound=true
      End If
Next

If Not (PackageFound AND ProgramFound AND CollectionFound) Then
       WScript.Echo "Either the package, program, or collection names were not
found"
       WScript.Echo "so the ad can't be created"
Else
       WScript.Echo "Creating advertisement: " & AdName

       'create a value for the time when the ad should be available to users
       formattedmonth= month(now)
       if len(formattedmonth)=1 then formattedmonth = "0" & formattedmonth
       formattedday= day(now)
       if len(formattedday)=1 then formattedday = "0" & formattedday
       datetime = year(now) & formattedmonth & formattedday &
left(formatdatetime(now, 4),2) & right(formatdatetime(now, 4),2) &
"00.000000+***"

       'create a value for the time when the ad should expire.
   'Default to + 1/2 year, but it's not enabled
       formattedmonth= month(DateAdd( "d",182, now))
    if len(formattedmonth)=1 then formattedmonth = "0" & formattedmonth
    formattedday= day(DateAdd( "d",182, now))
       if len(formattedday)=1 then formattedday = "0" & formattedday
  Set newAdvertisement = WbemServices.Get("SMS_Advertisement").SpawnInstance_()
   newAdvertisement.AdvertisementName = AdName
```

(continued)

Sample C.17 CreateAdvertisement.vbs — a script to create an advertisement *(continued)*

```
   newAdvertisement.comment = "created by NewAd.vbs"
       newAdvertisement.CollectionID = CollectionID
       newAdvertisement.PackageID = PackageID
       newAdvertisement.ProgramName = ProgramName
       newAdvertisement.PresentTime=datetime
       newAdvertisement.ExpirationTime= expdatetime
   newAdvertisement.Put_
End If
```

Modifying Advertisements

Sample C.18 changes an advertisement so that it either runs at the times it is assigned to run or ignores those times. Either way, because it is assigned, the users do not see it unless you select the **Allow users to run the program independently of assignments** property on the advertisement. The advertisement will only run after the assigned times and when the assignments are enabled.

This script can be useful if you have an advertisement that you want to force to run at night, but do not want users to run during the day. Usually, SMS does not allow you to set an advertisement to run that way. So, for example, you would create the advertisement with an assignment to run after 7:00 P.M. on the first day and then create a Windows NT Scheduler Task to run this script every evening at 7:00 P.M. and to run it again at 5:00 A.M. When using this script in a Windows NT Scheduler Task, you can cut or comment out the **WScript.Echo** lines.

Note that in this example the moniker, the WMI scripting model is used for simplicity.

Sample C.18 CreateCollection.vbs — a script to toggle an assigned advertisement

```
Set instance = GetObject(
"WinMgmts:root\SMS\site_B2k:SMS_Advertisement.AdvertisementID='B2K20001'")
WScript.Echo "Advertisement: " & instance.AdvertisementName
WScript.Echo "Assigned Schedule Enabled?: " & instance.AssignedScheduleEnabled
WScript.Echo ""

If instance.AssignedScheduleEnabled=True Then
       WScript.Echo "Was Enabled"
   instance.AssignedScheduleEnabled=False
       instance.Put_
    WScript.Echo "Now Disabled"
else
       WScript.Echo "Was Disabled"
   instance.AssignedScheduleEnabled=True
    instance.Put_
    WScript.Echo "Now Enabled"
end if
```

Unlocking Advertisements

If you have a site that used to be a child site but is now a central site, the site might have advertisements that originated at the old parent site. Those advertisements are locked because they would usually be managed from the parent site. But because there is no longer a parent site, you need to unlock them. This can be done with the **SMS_Advertisement Unlock** method.

Adding Assignments to an Advertisement

The dates and times that advertisements can be assigned to run are stored as an array of objects. The objects are based on one of the classes, which are based on the **SMS_ScheduleToken** class. Adding or removing assignments requires adding or removing objects from that array. Sample C.19 shows adding an assignment to an existing advertisement.

Because other properties in other SMS classes are also arrays of objects, the technique described in this section can be used in other situations.

Sample C.19 AdvertisementAddAssignment.vbs — a script to add an assignment to an advertisement

```
if wscript.arguments.count<>1 then
    WScript.Echo "One argument please - the advertisement ID"
    WScript.Quit
else
    AdID = wscript.arguments(0)
end if

Set loc = CreateObject("WbemScripting.SWbemLocator")
Set WbemServices = loc.ConnectServer( , "root\SMS\site_MSO")

Set Ad = WbemServices.Get("SMS_Advertisement.AdvertisementID='" & AdID & "'")

'copy the current assigned schedule array to a local array that is one
'bigger than the current assigned schedule array
dim array()
redim array( UBound(Aad.AssignedSchedule)+1 )
for i=0 to UBound( Ad.assignedSchedule )
  set array(i) = Ad.AssignedSchedule(i)
next

'create an assignment
Set newST = WbemServices.Get("SMS_ST_RecurInterval").SpawnInstance_()
newST.DayDuration=20
```

(continued)

Sample C.19 AdvertisementAddAssignment.vbs — a script to add an assignment to an advertisement *(continued)*

```
'add the new assignment to the end of the local array
set array( UBound(array) ) =newST

Ad.AssignedSchedule = array

Ad.Put_
```

Packages

Scripting the creation of packages can be useful for a variety of reasons, including:

- Converting many packages from the format that is used by another system (such as SMS 1.2) to the SMS package format.
- Adjusting package settings for many packages.
- Automating the distribution of packages to distribution points so that you can control their distribution beyond the controls offered by SMS addresses without having to have an administrator present.

Creating packages can be as simple as creating the package definition itself. However, packages are not useful until they have SMS programs that specify which programs should be run in the packages to accomplish certain functions (such as setup or deinstallation). Because packages that contain files also need to be distributed to distribution points, defining which distribution points should be used for a package can also be important.

Creating Packages and Programs

Creating a package object is the same as creating other SMS objects. The only required property is a name. SMS automatically fills in the details for many properties, including the package identifier, disconnection settings, priority, and action in progress. You can set properties that are significant to the task that your script is meant to accomplish, such as description, package source path, and package source flag. For more information about the **SMS_Package** class properties, see the *Microsoft Systems Management Server 2003 Software Development Kit*.

The **PkgSourceFlag** property is often important when creating packages. This property can have only one of the following values:

1. The package has no source files, which is the default.
2. The package has source files, and SMS should obtain them directly from the source path.
3. The package has source files, but SMS should make a compressed copy of them and then use that compressed copy.

Creating programs is also easy. Programs require a package identifier, program name, and command line. SMS automatically sets some properties, such as program flags. You can set other properties as needed, such as comment, estimated disk space, or run time. You can also set flags for how the program should run on the client. For more information about the **SMS_Program** class properties, see the *Microsoft Systems Management Server 2003 Software Development Kit*. Because the package comment is seen by end users, it should have a value that is meaningful to them.

Sample C.20 shows creating a typical package and program. Note that this package creates a package for Notepad.exe, which is presumed to be available in C:\SoftwareSource.

Sample C.20 CreatePackage.vbs — creates a package and a program

```
if wscript.arguments.count<>2 then
    WScript.Echo "Two arguments please: the package and program names"
    WScript.Quit
else
    PackageName    = wscript.arguments(0)
    ProgramName    = wscript.arguments(1)
end if

Dim loc
Set loc = CreateObject("WbemScripting.SWbemLocator")
Dim WbemServices
Set WbemServices = loc.ConnectServer( , "root\SMS\site_B2K")

Set newPackage = WbemServices.Get("SMS_Package").SpawnInstance_()
newPackage.Name = PackageName
newPackage.Description = "created by newpackage.vbs"
newPackage.PkgSourceFlag = 2
newPackage.PkgSourcePath = "C:\SoftwareSource"
path=newPackage.Put_

'and get the automatically assigned package ID
Set Package=wbemServices.Get(path)
PackageID= Package.PackageID

Set newProgram = WbemServices.Get("SMS_Program").SpawnInstance_()
newProgram.ProgramName = ProgramName
newProgram.PackageID = PackageID
newProgram.Comment = "phone the helpdesk for support with this program"
newProgram.CommandLine = "Notepad.exe"
newProgram.Put_
```

Sending Packages to Distribution Points

Packages with source files are useful only when they are sent to distribution points. For packages to be sent to distribution points, you must specify the distribution points to which you want the packages sent. You do this by creating objects of the **SMS_DistributionPoint** class.

Instances of the **SMS_DistributionPoint** class require a Network Abstraction Layer (NAL) path, a package ID, and the site code of the site at which the distribution points are located. You should also specify the site name. You can get the site code and NAL path by listing the available distribution points. You do this by querying the **SMS_SystemResourceList** class. You can get the site name from the site details of the **SMS_Site** class.

Sample C.21 makes a package available on all distribution points in your SMS hierarchy.

Sample C.21 AssignDistributionPoints — assigns all available distribution points to a package

```
if WScript.Arguments.Count<>1 then
    WScript.Echo "One argument please: the package ID"
       WScript.Quit
else
       PackageID = WScript.Arguments(0)
end if

Set loc = CreateObject("WbemScripting.SWbemLocator")
Dim WbemServices
Set WbemServices = loc.ConnectServer( , "root\SMS\site_B2K")

Set AllDPs = wbemServices.ExecQuery("Select * From SMS_SystemResourceList WHERE
RoleName='SMS Distribution Point'")
For Each DP In AllDPs
    Set Site = WbemServices.Get("SMS_Site='" & DP.SiteCode  & "'")

    Set newDP = WbemServices.Get("SMS_DistributionPoint").SpawnInstance_()
    newDP.ServerNALPath = DP.NALPath
    newDP.PackageID = PackageID
    newDP.SiteCode = DP.SiteCode
    newDP.SiteName = Site.SiteName
    newDP.Put_

Next
```

If you want to control when your package is first distributed to the distribution points (beyond the control that SMS addresses provides), you can run the script in Sample C.21 at that time. Do not assign distribution points to the package prior to that time. Run the script as a scheduled task with a suitably privileged account, if an administrator is not available at that time.

If you want to refresh specific distribution points for a package (possibly because they have been rebuilt) and you want to use a script (because you have many such distribution points or the timing is inconvenient), you can use the **RefreshNow** property of the **SMS_DistributionPoint** class, as shown in Sample C.22. If you want to refresh all the distribution points that are assigned to a package, you can use the **RefreshPkgSource** instance method of the **SMS_Package** class.

Sample C.22 RefreshDPs.vbs — part of a script that refreshes distribution points

```
Set DPs = wbemServices.ExecQuery("Select * From SMS_DistributionPoint WHERE
SiteCode='" & sitecode & "' AND PackageID='" & PackageID & "'")
For Each DP In DPs
   DP.RefreshNow = True
   DP.Put_
Next
```

The final detail that needs to be specified when creating packages is access accounts. By default, administrators get full control, and guests and users get read access to every package, even when the package is created by using a script. Access accounts can be controlled from your script by using the **SMS_PackageAccessByUsers** class.

Security Rights

SMS object security is used to control which kinds of objects (such as collections or packages) and which particular objects (specific collections or packages) that SMS administrators can manage. The kinds of objects that can be managed are controlled with class-level rights, and the particular objects are controlled with instance-level rights.

Four SMS classes can be used with SMS object security, but only the first class is commonly used in SMS scripting:

SMS_UserInstancePermissions The rights for particular SMS objects. Each instance in this class represents all the rights for a particular user or group on a particular instance.

SMS_UserInstancePermissionNames The rights for particular SMS objects. Each instance in this class represents a particular right for a particular user or group on a particular instance. This class is useful for display purposes because the rights are already parsed into a form that you can read, but it takes more scripting to achieve the same results as the **SMS_UserInstancePermissions** class.

SMS_UserClassPermissions The rights for kinds of objects. Each instance in this class represents all the rights for a particular user or group for a particular class.

SMS_UserClassPermissionNames The rights for kinds of objects. Each instance in this class represents a particular right for a particular user or group for a particular class. The class is useful for display purposes because the rights are already parsed into a form that you can read, but it takes more scripting to achieve the same results as the **SMS_UserClassPermissions** class.

Sample C.23 grants read rights and modify rights to a user group for all collections at the instance level. Adding the same rights at the class level is easier, but this allows administrators to do tasks that are not intended.

Sample C.23 GrantRights.vbs — grants rights to a user group for all collections

```
sitecode="B2K"
SMSJuniorAdmins="BEIGE\SMS Junior Admins"

Dim lLocator
Set lLocator = CreateObject("WbemScripting.SWbemLocator")
Dim WbemServices
Set WbemServices = lLocator.ConnectServer( , "root\sms\site_" & sitecode)

Set Collections = WbemServices.ExecQuery("Select * From SMS_Collection")
For Each Collection In Collections
    'ignore this special collection
    If (Collection.CollectionID <> "COLLROOT") AND Collection.OwnedByThisSite
Then
       WScript.Echo vblf & Collection.Name & "  " & Collection.CollectionID

       AlreadySet = False
       Set Rights = WbemServices.ExecQuery("Select * From
SMS_UserInstancePermissionNames WHERE ObjectKey=1 AND InstanceKey='" &
Collection.CollectionID & "'")
       For Each Rite in Rights
           WScript.Echo "  " & Rite.Username & "  " & Rite.PermissionName
        If Rite.Username = SMSJuniorAdmins Then AlreadySet=True
       Next

       If Not AlreadySet Then
          Set newRight =
WbemServices.Get("SMS_UserInstancePermissions").SpawnInstance_()
          newRight.UserName = SMSJuniorAdmins
          newRight.ObjectKey = 1 'collections
          newRight.InstanceKey = Collection.CollectionID
          newRight.InstancePermissions = 1+2 'just Read & Modify
          newRight.Put_
      WScript.Echo "  The junior administrators now have read and modify access
to this collection."
       End If
    End If
Next
```

Working with SMS Site Settings

Working with SMS site settings in a script can be useful to:

- Change a site setting in a large number of sites without human intervention.
- Report on site settings (to confirm all sites are set up correctly).

SMS is controlled through site control files. Site control files are represented in WMI by using the **SMS_SiteControlFile** class. **SMS_SiteControlFile** has methods to enable scripts to manipulate site control files. The methods allow your scripts to retrieve a current copy of a site control file and to later commit any changes.

Important

For security reasons, SMS account passwords cannot be set in scripts. You can, however, script SMS Administrator console operations to simulate an administrator setting the passwords. For more information, see the "Scripting Console Operations" section later in this appendix.

How to Work with SMS Site Settings

Working with SMS site settings in scripts involves the following steps:

1. Create a session handle to use with site control files.
2. Get a site control file for the site that you want to manage. Site control files are available for the site to which your script connects and for all sites that report to that site. You must also specify whether you want to use file type 2 (proposed settings — to change settings) or type 1 (actual settings — to report on settings). You can directly change file type 1; however, if you change file type 1, the change will not get processed by Hierarchy Manager and Site Component Manager, and thus will not have any effect.
3. Get a class based on the **SMS_SiteControlItem** class that represents the relevant part of the site control file that you need. If a new item is being created, such as an address, this involves launching an instance of the class. Usually, your script gets a specific instance of the class. For more information about the available **SMS_SiteControlItem** classes, see the *Microsoft Systems Management Server 2003 Software Development Kit*. Also, read a sample site control file to find the details that you plan to change, and then determine the class based on the group that the details are stored in.
4. Change the values.
5. Save the changes.
6. Commit the site control file changes.
7. Release the session handle.

Steps 1, 2, 5, 6, and 7 do not change much, if at all, from script to script. Therefore, you can copy them from any sample script that works with SMS site settings.

Step 3 varies in limited ways. If you cannot find an sample script that is similar to what you need, you might be able to piece the details together with some experimentation. What is important is that you select the correct **SMS_SiteControlItem** class and that you specify valid values for the key properties that are used to retrieve an instance of the class.

As with other scripts, it might be useful to look at the available instances of the class. Browsing instances can be done with CIM Studio or similar tools, but for **SMS_SiteControlItem** classes you cannot simply use the **Instances** button. Instead, you must execute a query, by using the **WQL Queries** button, such as shown in the following:

```
Select * FROM SMS_SCI_ClientComp WHERE SiteCode="B2K" AND FileType=1
```

Sample C.24 is a typical script for working with SMS site settings, except that the details for step 4 are not illustrated. For examples of step 4, see the other samples in this appendix.

Sample C.24 SiteSettings.vbs — a commented template of a typical site setting script

```
strSiteCode="MSO"
Set lLocator = CreateObject("WbemScripting.SWbemLocator")
Set wbemservices = lLocator.ConnectServer( , "root\sms\site_" & strSiteCode)

'step 1:
set WbemContext=CreateObject("WbemScripting.SWbemNamedValueSet")
WbemContext.Add
"SessionHandle",WbemServices.ExecMethod("SMS_SiteControlFile","GetSessionHandle"
).SessionHandle
'step 2:
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=2,Sitecode=""" &
strSiteCode & """", "Refresh", , , WbemContext
'step 3 (details will vary, and sometimes an instance will be spawned, rather
'than retrieved):
Set WbemInst = WbemServices.Get("SMS_SCI_ClientComp.Filetype=1,Itemtype=""Client
Component"",Sitecode=""" & Sitecode & """,ItemName=""Remote Control""", ,
WbemContext)
'step 4 varies widely
'step 5:
WbemInst.Put_ , WbemContext
'step 6:
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=1,Sitecode=""" &
strSitecode & """", "Commit", , , WbemContext
'step 7:
WbemServices.Get("SMS_SiteControlFile").ReleaseSessionHandle
WbemContext.Item("SessionHandle").Value
```

Step 4 can be challenging if you do not have a good sample script. This can be complex due to the variety of properties that can be involved, such as simple properties, objects, arrays, and arrays of objects. Writing the script in Visual Basic (as described in the "Scripting in Visual Basic" section earlier in this appendix) and using the **Locals** window can help verify details.

Reporting Site Component Settings

Sample C.25 shows the site boundaries for all sites that are reporting to the site that the script is run against, including the site itself.

Sample C.25 Boundaries.vbs — lists all the site boundaries for all sites

```
Sub Level( parent, depth )

    Set Sites = WbemServices.ExecQuery("Select * From SMS_Site Where
ReportingSiteCode='"+parent+"'")
    For Each Site In Sites

    'descriptive info for site
    Descr=Left( SPACE( depth*3 ) & Site.SiteName + " ("+Site.SiteCode+") ", 29)
    Descr=Descr & SPACE(30-LEN(Descr))
    wscript.echo descr

        set boundaries =
WbemServices.Get("SMS_SCI_SiteAssignment.SiteCode='"+site.sitecode+"',Filetype=1
,ItemName='Site Assignment',ItemType='Site Assignment'")
    For i=0 to ubound(boundaries.AssignDetails)
           Wscript.Echo space(len(descr)) & boundaries.AssignTypes(i) & ": " &
boundaries.AssignDetails(i)
    next

    Level Site.SiteCode, depth+1

    Next

End Sub

    Sitecode = "NES"

    Set lLocator =  CreateObject("WbemScripting.SWbemLocator")
    Set WbemServices = lLocator.ConnectServer( , "root\sms\site_" & sitecode )

    Level "", 0
```

Adjusting Component Settings

Sample C.26 changes a configuration parameter for the Site Component Manager.

Sample C.26 SiteComp.vbs

```
Sitecode = "B2K"
Server = "BEIGE-W2K-SRVR"

Set lLocator =  CreateObject("WbemScripting.SWbemLocator")
Set WbemServices = lLocator.ConnectServer(, "root\sms\site_B2K")
```

(continued)

Sample C.26 SiteComp.vbs *(continued)*

```
'connect to the site control file
Set WbemContext=CreateObject("WbemScripting.SWbemNamedValueSet")
WbemContext.Add "SessionHandle", WbemServices.ExecMethod("SMS_SiteControlFile",
GetSessionHandle").SessionHandle

'Refresh our copy of the SiteControlFile
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=1,Sitecode=""" & Sitecode
& """", "Refresh", , , WbemContext

'Retrieve Site Control Item instance
Set WbemInst =
WbemServices.Get("SMS_SCI_Component.Filetype=1,Itemtype=""Component"",Sitecode="
"" & Sitecode & """,ItemName=""SMS_SITE_COMPONENT_MANAGER|" & UCase(Server) &
""""", , WbemContext)

'Retrieve the 'props' property
PropArray = WbemInst.props
x = PropArray(0).Value
Wscript.echo x
PropArray(0).Value = x + 1
WbemInst.props = PropArray

'Store the instance
WbemInst.Put_ , WbemContext

'Execute SMS_SiteControlFile::Commit to update the Site Control file with
'the new InterPoll Delay time
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=1,Sitecode=""" & Sitecode
& """", "Commit", , , WbemContext

'release the site control file
WbemServices.Get("SMS_SiteControlFile").ReleaseSessionHandle
WbemContext.Item("SessionHandle").Value
```

Setting the Site Comment for a Secondary Site

Sample C.27 shows how a script that manipulates site settings at one site can change the properties for another site that reports to the first site, even if the other site reports indirectly to the first site. The ConnectServer line connects to the parent site, but the line that refreshes the script's copy of the site control file and the line that gets the site definition both get the child site's site control file. When the changes are committed, the child site is again specified.

Sample C.27 SiteComment.vbs

```
localSitecode = "B2K"
sitetochange = "BSE"

Set lLocator = CreateObject("WbemScripting.SWbemLocator")
Set WbemServices = lLocator.ConnectServer( , "root\sms\site_" & localsitecode )

Set WbemContext = CreateObject("WbemScripting.SWbemNamedValueSet")
WbemContext.Add "SessionHandle", WbemServices.ExecMethod("SMS_SiteControlFile",
"GetSessionHandle").SessionHandle
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=1,Sitecode=""" &
Sitetochange & """", "Refresh", , , WbemContext
Set WbemInst =
WbemServices.Get("SMS_SCI_SiteDefinition.Filetype=1,Itemtype='Site
Definition',Sitecode='" & Sitetochange & "',ItemName='Site Definition'", ,
WbemContext)

proparray = WbemInst.props
WScript.Echo sitecode & " site comment: " & proparray(0).Value1
proparray(0).Value1 = "set from script"
WbemInst.props = proparray

WbemInst.Put_ , WbemContext
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=1,Sitecode=""" &
sitetochange & """", "Commit", , , WbemContext
WbemServices.Get("SMS_SiteControlFile").ReleaseSessionHandle
WbemContext.Item("SessionHandle").Value
```

Embedding Properties

Some site settings have objects embedded within their properties, often as an array of objects; for example, addresses and site status rules. Some SMS objects also have embedded objects; for example, the **AssignmentSchedule** property of advertisements contains **SMS_ScheduleToken** objects.

When an object requires embedded objects, your script must create the embedded objects as usual. (You launch an instance and set the properties.) Then, instead of saving the object (with a **Put** method), the embedded object is assigned to the property or to an element of the property array, if the property is an array. When the main object is saved, the embedded objects are also saved within the main object.

Sample C.28 shows adding embedded objects to an address; for example, by adding the connect point. Sample C.19 shows adding embedded objects (assignments) to an advertisement. And Sample C.31 shows adding embedded objects (status filter rules) to a site component.

Creating Addresses

Creating SMS addresses is a typical need for scripting SMS site settings. If you set up many sites, you probably have some sites that have many sites reporting to them. The parent sites require an address for each child site. Sample C.28 shows how you can create an address in a script. With some minor changes, you can then have it input the relevant details from a spreadsheet, text file, or other source.

The difficult part of creating addresses in a script is that you cannot set the user name and password for the SMS Site Address account. SMS account passwords cannot be set in scripts. However, you can script SMS Administrator console operations to simulate an administrator setting the passwords. For more information, see the "Scripting Console Operations" section later in this chapter.

Sample C.28 CreateAddress.vbs — creates SMS addresses

```
strSiteCode="MSO"
Set lLocator = CreateObject("WbemScripting.SWbemLocator")
Set wbemservices = lLocator.ConnectServer( , "root\sms\site_" & strSiteCode)

set WbemContext=CreateObject("WbemScripting.SWbemNamedValueSet")
WbemContext.Add
"SessionHandle",WbemServices.ExecMethod("SMS_SiteControlFile","GetSessionHandle"
).SessionHandle
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=2,Sitecode=""" &
strSiteCode & """", "Refresh", , , WbemContext

'Spawn an SMS_SCI_Address instance
Set WbemInst = WbemServices.Get("SMS_SCI_Address").SpawnInstance_()

wbeminst.AddressType = "MS_LAN"
wbeminst.DesSiteCode = "PS1"
wbeminst.ItemName = "PS1|MS_LAN"
wbeminst.ItemType = "Address"
wbeminst.Order = 1
wbeminst.SiteCode = strSitecode
wbeminst.UnlimitedRateForAll = 1

Set newep1 = WbemServices.Get("SMS_EmbeddedProperty").SpawnInstance_()
newep1.PropertyName = "Connection Point"
newep1.Value = 0
newep1.Value1 = "RBSCS503"
newep1.Value2 = "SMS_SITE"

Set newep2 = WbemServices.Get("SMS_EmbeddedProperty").SpawnInstance_()
newep2.PropertyName = "LAN Login"
newep2.Value = 0
```

(continued)

Sample C.28 CreateAddress.vbs — creates SMS addresses *(continued)*

```
newep2.Value1 = "username"  'there's no way to calculate this value
newep2.Value2 = "password"  'there's no way to calculate this value

wbeminst.Props = Array(newep1, newep2)

WbemInst.Put_ , WbemContext
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=1,Sitecode=""" &
strSitecode & """", "Commit", , , WbemContext
WbemServices.Get("SMS_SiteControlFile").ReleaseSessionHandle
WbemContext.Item("SessionHandle").Value
```

Adjusting Client Agent Settings

In scripts, you manage client agent settings in much the same way that you manage component settings. The primary difference is that you use a different **SMS_SiteControlItem** class:

```
'Retrieve Site Control Item instance
Set WbemInst = WbemServices.Get("SMS_SCI_ClientComp.Filetype=1,Itemtype=""Client
Component"",Sitecode=""" & Sitecode & """,ItemName=""Remote Control""", ,
WbemContext)
```

Sample C.29 sets all the values for Remote Tools. This includes enabling the client agent.

Sample C.29 RemoteControl.vbs — enables and configures remote control

```
Sitecode = "WMA"
Set lLocator =  CreateObject("WbemScripting.SWbemLocator")
Set WbemServices = lLocator.ConnectServer( , "root\sms\site_" & sitecode )

'connect to the site control file
Set WbemContext=CreateObject("WbemScripting.SWbemNamedValueSet")
WbemContext.Add "SessionHandle", WbemServices.ExecMethod("SMS_SiteControlFile",
"GetSessionHandle").SessionHandle
'Refresh our copy of the SiteControlFile
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=0,Sitecode=""" & Sitecode
& """", "Refresh", , , WbemContext

'Retrieve Site Control Item instance
Set WbemInst =
WbemServices.Get("SMS_SCI_ClientComp.Filetype=1,Itemtype=""Client
Component"",Sitecode=""" & strSiteCode & """,ItemName=""Remote Control""", ,
WbemContext)

'Enable the Remote Tools Client Agent
WbemInst.Flags = 1
```

(continued)

Sample C.29 RemoteControl.vbs — enables and configures remote control *(continued)*

```
'Retrieve the 'props' property.
arrProps = WbemInst.props

'Do not allow Chat (default for Policy: Full level of remote access allowed)
arrProps(0).Value = 0
'Do not allow users to change remote control settings
arrProps(1).Value = 0
'Allow file transfer (default for Policy: Full level of remote access allowed)
arrProps(2).Value = 1
'Allow reboot (default for Policy: Full level of remote access allowed)
arrProps(3).Value = 1
'Allow remote execute (default for Policy: Full level of remote access allowed)
arrProps(4).Value = 1
'Allow takeover (default for Policy: Full level of remote access allowed)
arrProps(5).Value = 1
'Allow view configuration (default for Policy: Full level of remote access
allowed)
arrProps(6).Value = 1
'Not always visible (Notification Tab: 1 = Also show indicator when no session
is active)
arrProps(7).Value = 0
'Audible signal (Notification Tab: Play A Sound(0 = No, 1 = Repeatedly during
'                                                  session, 2 = When session
begins and ends)
arrProps(8).Value = 2
'Compression type (Advanced Tab: 0 = Low(RLE), 1 = High(LZ)
'                                2 = Automatically Select)
arrProps(9).Value = 2
'Control level (Policy Tab: Level of remote access allowed (0 = None,
'                                                 1 = Limited, 2 = Full)
arrProps(10).Value = 2
'Default protocal (Advanced Tab: 0 = TCP/IP, 1 = NetBIOS, 2 = IPX)
arrProps(11).Value = 0
'Indicator type (Notification Tab: 0 = Show status icon on taskbar,
'                             1 = Show high-security indicator on desktop)
arrProps(12).Value = 0

'Lana Number (Advanced Tab: 0 = Default for TCP/IP)
arrProps(13).Value = 0
'Permission required (Policy Tab: 0 = Do not ask for permission,
'                                 1 = Display message to ask for permission)
arrProps(15).Value = 0
'Use IDIS (Advanced Tab: 1 = Install accelerated screen transfer on Windows NT
Clients)
arrProps(16).Value = 1
```

(continued)

Sample C.29 RemoteControl.vbs — enables and configures remote control *(continued)*

```
'Visible signal (Notification Tab:  1 = Display a visual indicator)
arrProps(17).Value = 1

WbemInst.props = arrProps

'Prepare Remote Control Permitted Viewers Table
Set RegMultStr = WbemServices.Get( "SMS_Client_Reg_MultiString_List"
).SpawnInstance_()
RegMultStr.ValueName = "PermittedViewers"
RegMultStr.ValueStrings = array(arrValueStrings)
RegMultStr.ValueStrings(0) = "Administrators"
RegMultStr.ValueStrings(1) = "MYDOMAIN\SMS MMC Admin"
RegMultStr.ValueStrings(2) = "MYDOMAIN\SMS Roll Out"
RegMultStr.ValueStrings(3) = "MYDOMAIN\SMS Remote Control"

WbemInst.RegMultiStringLists(0).ValueName = RegMultStr.ValueName
WbemInst.RegMultiStringLists(0).ValueStrings = RegMultStr.ValueStrings

'Store the instance
WbemInst.Put_ , WbemContext

'Execute SMS_SiteControlFile::Commit to update the Site Control file
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=0,Sitecode=""" & Sitecode
& """", "Commit", , , WbemContext

'release the site control file
WbemServices.Get("SMS_SiteControlFile").ReleaseSessionHandle
WbemContext.Item("SessionHandle").Value
```

Adding Boundaries

Sample C.30 reads boundaries from an Excel spreadsheet and adds them to the site.

Sample C.30 AddBoundaries.vbs — adds boundaries to a site

```
Sitecode = "MSO"

Set lLocator =  CreateObject("WbemScripting.SWbemLocator")
Set WbemServices = lLocator.ConnectServer( , "root\sms\site_" & sitecode )
Set WbemContext=CreateObject("WbemScripting.SWbemNamedValueSet")
WbemContext.Add "SessionHandle", WbemServices.ExecMethod("SMS_SiteControlFile",
"GetSessionHandle").SessionHandle
```

(continued)

Sample C.30 AddBoundaries.vbs — adds boundaries to a site *(continued)*

```
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=1,Sitecode='" & Sitecode &
"'", "Refresh", , , WbemContext
Set WbemInst =
WbemServices.Get("SMS_SCI_SiteAssignment.Filetype=2,Itemtype='Site
Assignment',Sitecode='" & Sitecode & "',ItemName='Site Assignment'", ,
WbemContext)

'Retrieve the boundary details
proparray1 = WbemInst.AssignDetails
proparray2 = WbemInst.AssignTypes

set Excel = CreateObject( "Excel.Application" )
Set wkbooks = Excel.Workbooks
wkbooks.Open( "c:\Scripts\Newest\Boundaries.xls" )

i=3
boundary="x"
While boundary<>""

   boundary = wkbooks.Item(1).ActiveSheet.Cells(i, 1)
   boundary_type = wkbooks.Item(1).ActiveSheet.Cells(i, 2)

   if boundary<>"" Then
      i=i+1

      onemore = ubound(proparray1) + 1
      redim preserve proparray1( onemore )
      redim preserve proparray2( onemore )

      proparray1( onemore ) = boundary
      proparray2( onemore ) = boundary_type

   end if


Wend

Excel.Quit

WbemInst.AssignDetails = proparray1
WbemInst.AssignTypes = proparray2
WbemInst.Put_ , WbemContext

WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=0,Sitecode=""" & Sitecode
& """", "Commit", , , WbemContext
WbemServices.Get("SMS_SiteControlFile").ReleaseSessionHandle
WbemContext.Item("SessionHandle").Value
```

Creating Site Systems

Creating site systems such as client access points (CAPs), distribution points, and management points is done by creating instances in the **SMS_SCI_SysResUse** class. A Network Abstraction Layer (NAL) path that uniquely identifies the site system is required for the **NALPath** and **ItemName** properties of the **SMS_SCI_SysResUse** instances. **NALPath** properties are created by using the **PackNALPath** method of the **SMS_NAL_Methods** class. The NAL path is also needed for the **Objects Polled By Site Status** embedded object in the **PropLists** property for all site systems and for the **InboxRoot** embedded object in the **Props** property for CAPs. Other embedded objects are required in the **Props** property for other site systems.

You should create site systems by using the SMS Administrator console and observe the properties that are set for those site systems. You can then reproduce the same actions in your script.

Managing Status Filter Rules

Status filter rules might be the most complex SMS elements to manage in scripts. Managing status filter rules requires manipulation of arrays that are embedded in an array of objects that is itself a property of an object. Sample C.31 shows how to create status filter rules from a script.

Sample C.31 creates a status filter rule that causes all other status filter rules to not be evaluated if status message 4711 is received. Nothing is done with the message 4711 itself. Status message 4711 indicates that the SMS Site System Status Summarizer has detected that a storage object has less free storage space than the critical free space threshold. As a result, this rule ensures that the remaining space is not consumed with status messages.

Sample C.31 StatusFilterRule.vbs

```
Const strSiteCode = "B2K"          ' SMS Sitecode
Const strServer = "BEIGE-W2K-SRVR" ' SMS Site Server
Set lLocator = CreateObject("WbemScripting.SWbemLocator")
Set WbemServices = lLocator.ConnectServer(, "root\sms\site_" & strSiteCode)

Set WbemContext=CreateObject("WbemScripting.SWbemNamedValueSet")
WbemContext.Add "SessionHandle", WbemServices.ExecMethod ("SMS_SiteControlFile",
"GetSessionHandle").SessionHandle
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=1,Sitecode=""" &
strSitecode & """", "Refresh", , , WbemContext
myObjPath ="SMS_SCI_Component.FileType=1,Itemtype=""Component"",Sitecode=""" & _
                          strSiteCode & """,ItemName=""SMS_STATUS_MANAGER|" &
ucase(strServer) & """"
Set WbemInst = WbemServices.Get(myObjPath, , WbemContext)
PropLists = WbemInst.PropLists
```

(continued)

Sample C.31 StatusFilterRule.vbs *(continued)*

```
'the rule details!
myFilterRuleName = "My new filter rule"
myFilterRuleArray = array( "Actions=0", "Actions Mask=16", "Stop Evaluating=1",
"Code=4711" )

'check if the rule is already present
Rc = 0
For i = 0 to ubound(PropLists)
    If PropLists(i).PropertyListName = "Status Filter: SMS_STATUS_MANAGER" Then
                  intPointer1 = i                                          ' Store
the index, we need him later
               Rc = 1                                            ' Set Rc to
indicate that the object was found
               For j = 0 to ubound(PropLists(i).Values)  ' Search for the filter
rule
                        If PropLists(i).Values(j) = myFilterRuleName Then
                                          Rc = 2                          ' Set
Rc to indicate that the status filter rule was found
                        End If
               Next
    End If
Next
If Not (Rc = 1) Then
       WScript.Echo "Rule has been added already or we couldn't find ""Status
Filter: SMS_STATUS_MANAGER"""
       WScript.Quit
End If

Set embprops = WbemInst.Properties_("PropLists")

'add the name of the new rule to the "Status Filter: SMS_STATUS_MANAGER" object
tmparray = embprops.Value(intPointer1).Values              'copy the array (all
filter rules)
redim preserve tmparray( ubound(tmparray) + 1 )           'extend array by 1
element
tmparray( ubound(tmparray) ) = myFilterRuleName          'add the new element
embprops.Value(intPointer1).Values =  tmparray            'overwrite the array
(all filter rules)

'build an embedded property
Set ObjEmb = WbemServices.Get("SMS_EmbeddedPropertyList")
Set objEmbInst = ObjEmb.SpawnInstance_()
objEmbInst.PropertyListName = "Status Filter Rule: " & myFilterRuleName
objEmbInst.Values = myFilterRuleArray
```

(continued)

Sample C.31 StatusFilterRule.vbs *(continued)*

```
'add the rule itself (the embedded property) to the list of rules
embprops.Value(Ubound(embprops.value) + 1) = objEmbInst

WbemInst.Put_ , WbemContext

'Execute SMS_SiteControlFile::Commit to update the Site Control file
WbemServices.ExecMethod "SMS_SiteControlFile.Filetype=1,Sitecode=""" &
strSitecode & """", "Commit", , , WbemContext
WbemServices.Get("SMS_SiteControlFile").ReleaseSessionHandle
WbemContext.Item("SessionHandle").Value
WbemContext.Remove "SessionHandle"
WScript.DisconnectObject WbemContext
WScript.DisconnectObject lLocator
```

The unique parts of Sample C.31 are the array manipulation and the line that sets the status filter rule array:

```
myFilterRuleArray = array( "Actions=0", "Actions Mask=16", "Stop Evaluating=1",
"Code=4711" )
```

Many details can be set in the status filter rules array, as listed in Table C.2.

Table C.2 Possible Status Filter Rule Array Details

Detail	Description	Possible value	Note
Actions	Flags for the actions to be carried out on the rules	See the next table	Required
Actions Mask	Used with the actions property	See the next table	Required
Stop Evaluating	Do not process lower priority status filter rules	Boolean (1=TRUE, 0=FALSE)	
Days to Keep	The number of days to keep messages if they are written to the SMS database	An integer	Only valid if the Actions and Actions Mask values are set to write the message to the SMS database
Program	The program to run if a program is to be run	A string	Only valid if the Actions and Actions Mask values are set to run a program
Module Name	Source — client, server, or the SMS Provider	See a status filter rule properties dialog box in the SMS Administrator console for the possible values	

(continued)

Table C.2 Possible Status Filter Rule Array Details *(continued)*

Detail	Description	Possible value	Note
Site Code	Site code	A string	
This Site	Used when the site code is the current site	Boolean (1=TRUE, 0=FALSE)	
Machine Name	System – computer name	A string	
Component	Component	See a status filter rule properties dialog box in the SMS Administrator console for the possible values	
Message Type	Message type – Milestone, Detail, or Audit	256, 512, or 768, respectively	
Severity	Severity – Informational, Warning, or Error	1073741824, 2147483648, or 3221225472, respectively	
Code	Message ID	A valid SMS status message ID	
Attribute ID	Property; meaning object ID or resource ID. Different attributes IDs are relevant depending on the module name (source) selected. Possible attribute IDs are Package Id, Advertisement ID, Collection ID, User Name, and Distribution Point, respectively.	400, 401, 402, 403, or 404, respectively.	
Attribute Value	Property value, but means object ID value	An SMS object ID	

The Actions and Actions Mask flags in the status filter rule must be set to the values listed in Table C.3. You can set multiple values from this table to enable multiple actions, in which case the values should be added together.

For example, if you want to forward a message to the summarizers and replicate it to the parent site at medium priority, you would set the Actions value to 24 (16 + 8), and the Actions Mask value to 12. If you only want to write the message to the SMS database, Actions would be set 1 and Actions Mask to 17 (1+16).

Table C.3 Actions and Actions Mask Flags

Task	Actions flag	Actions Mask flag	Notes
Write the message to the SMS database	1	1	Both these flags must be set, if either is set
Report the message to the Windows event log	2	2	Both these flags must be set, if either is set
Whether to the message replicate to the parent site, and priority if done	4/8/12 - low, medium, or high, respectively (replication priority)	12 (replicate)	Both these flags must be set, if either is set
Whether to forward the message to the status summarizers	16 (the message should be forwarded)	16 (the message should not be forwarded)	Only one of these flags must be set
Run a program	32	32	Both these flags must be set, if either is set

Scripting Console Operations

In some cases, creating a script to simulate SMS operations that are usually done by using the SMS Administrator console might be difficult or even impossible. For example, conventional SMS scripts cannot set account details. In such cases, an alternative is to script the SMS Administrator console itself.

Microsoft Management Console (MMC) 2.0 includes a scripting object model. This object model can be used to control most of the SMS Administrator console functions. Any functions that cannot be controlled by the MMC 2.0 scripting object model can be controlled by using the WSH **SendKeys** object, which simulates keyboard entries. If the function can be controlled from the keyboard, **SendKeys** can control it. **SendKeys** can be used to control most of the SMS Administrator console, but the MMC 2.0 scripting object model is generally more reliable. If anything causes the Windows focus to be shifted from the SMS Administrator console, **SendKeys** can be directed to the wrong application.

MMC 2.0 is including with Windows XP and the operating systems in the Windows Server 2003 family. You must run MMC 2.0-based scripts from an SMS Administrator console on a computer running Windows XP or the operating systems in the Windows Server 2003 family.

Sample C.32 shows using the MMC 2.0 scripting object model and **SendKeys** to control the SMS Administrator console. Sample C.32 opens the SMS Administrator console to the site's addresses and then sets them all to use the same predefined account and password. For obvious security reasons, you need to ensure that only authorized staff can read any script that contains account details.

Sample C.32 SetAddressAccount.vbs — sets address security details

```
Set mmc = WScript.CreateObject( "MMC20.Application.1" )
mmc.Load ("\sms\bin\i386\sms.msc")
mmc.Show
Set doc = mmc.Document
Set namespace = doc.ScopeNamespace
Set View = doc.Views.Item(1)
mmc.UserControl = 1   'so the console doesn't close prematurely
set WshShell = WScript.CreateObject("WScript.Shell")

Set sms = namespace.GetRoot
Set Database = namespace.GetChild(sms)
While Left(Database.Name, 7) = "Connect": WScript.Sleep 1000: Wend
namespace.Expand (Database)
WScript.Sleep 2000
Set extra = namespace.GetChild(Database)
namespace.Expand (extra)
WScript.Sleep 6000
Set hierarchy = namespace.GetChild(extra)
namespace.Expand (hierarchy)
WScript.Sleep 8000
Set site = namespace.GetChild(hierarchy)
namespace.Expand (site)
WScript.Sleep 2000
Set settings = namespace.GetChild(site)
namespace.Expand (settings)
WScript.Sleep 2000
Set address = namespace.GetChild(settings)

View.ActiveScopeNode = address
WScript.Sleep 5000

'for all addresses in the console
for i=1 to View.ListItems.Count

   'next address
   Set objNode1 = View.ListItems(i)
   View.Select objNode1

   SetAddressAccount  'they're all set to the same username and password

next

WScript.Sleep 1000
mmc.quit
WScript.Sleep 500
```

(continued)

Sample C.32 SetAddressAccount.vbs — sets address security details *(continued)*

```
WshShell.SendKeys "N"

'-----------------------------------------------------------

Sub SetAddressAccount

View.ExecuteSelectionMenuItem ("Properties")

WScript.Sleep 500
WshShell.SendKeys "%E"                    'alt E - Set... button
WScript.Sleep 500
WshShell.SendKeys "testdomain\testuser"   'username
WScript.Sleep 500
WshShell.SendKeys "{TAB}"
WScript.Sleep 500
WshShell.SendKeys "password"                      'password
WScript.Sleep 500
WshShell.SendKeys "{TAB}"
WScript.Sleep 500
WshShell.SendKeys "password"                      'password again
WScript.Sleep 500
WshShell.SendKeys "{ENTER}"                       'OK password dialog box done
WScript.Sleep 500
WshShell.SendKeys "{ENTER}"                       'OK address dialog box done

End Sub
```

Scripting Client Operations

Scripting SMS client operations can be useful in a variety of situations, including:

- Creating discovery data records (DDRs) for clients to discover clients using unconventional techniques and possibly to force their installation (with Client Push Installation).
- Creating status Management Information Format (MIF) files to extend advertisement status reporting for software distributions.
- Extending hardware inventory to collect details about clients that are impossible or inefficient to collect without the flexibility of scripting.
- Configuring or controlling Advanced Client activities or presenting users with a customized interface.

Other than the hardware inventory extensions, scripting client operations does not use WMI. In all cases, scripting client operations does not use the same specific techniques that have been discussed so far in this appendix, although the general techniques of scripting are still similar.

Creating DDRs for clients

Creating DDRs for clients allows you to discover clients from any source that you might have available. When the clients are discovered, their installation can be forced with Client Push Installation. To create DDRs from scripts, use the SMS Resource Generator Object, which is included with the Active Directory/SMS Synchronization Tools. This tool is available from the Microsoft Web site at http://www.microsoft.com/smserver.

Keep in mind the following points when using the SMS Resource Generator Object:

- The SMS Resource Generator Object can be used on an SMS site server or SMS Legacy Clients. The SMS Resource Generator Object files must be installed on any computer that runs SMS Resource Generator Object scripts. You can distribute these files with an SMS package that installs the Active Directory/SMS Synchronization Tools with a command line of `SMSAdsync.exe /s`. To create the SMS Resource Generator Object Visual Basic programs, add a reference to "SMSRsGenCtl:SMS Resource Generator," and then DIM DDR as a new SMSResGen.
- SMS Resource Generator Object does not have to be run on the clients that are to be discovered. The DDRs can be created on any SMS computer.

Sample C.33 shows creating a DDR. This sample can be extended to read in the required details from a spreadsheet, Active Directory, or any other source. To create a completely accurate DDR for the computer on which the script is run, additions would have to be made to collect TCP/IP addresses, subnets, and media access control (MAC) addresses.

Sample C.33 CreateDDR.vbs – creates a DDR

```
newresource=0  'set to 1 if you want to create a new computer DDR, rather than
generate a DDR for the computer the script is run on

Const ADDPPROP_NONE = &H0
Const ADDPROP_GUID = &H2
Const ADDPROP_KEY = &H8
Const ADDPROP_NAME = &H44

'get the current GUID, if available
Set WshSHell=WScript.CreateObject("WScript.Shell")
On Error Resume Next
Version=WSHShell.RegRead("HKEY_LOCAL_MACHINE\Software\Microsoft\SMS\Client\Clien
t Components\SMS Client Base Components\Installation Properties\Installed
Version")
if err=0 AND newresource<>1 then
    On Error Goto 0
    if Version="99.9.9999.9999" Then  'it's a Advanced Client, so get the GUID
from WMI
```

(continued)

Sample C.33 CreateDDR.vbs – creates a DDR *(continued)*

```
                    GUID=GetObject("WinMgmts:root\CCM:CCM_Client=@").ClientID
    else                              'it's a Legacy Client or 2.0 client
           GUID=WSHShell.RegRead(
"HKEY_LOCAL_MACHINE\Software\Microsoft\SMS\Client\Configuration\Client
Properties\SMS Unique Identifier")
       end if
else                                  'it's not a client yet, so create a new GUID
       On Error Goto 0
    GUID=CreateObject("Scriptlet.TypeLib").GUID
end if
On Error Goto 0

if newresource=1 then
    Randomize
    Computer="TestComputer" & Int(Rnd()*10000)
else
      Set WshNetwork=WScript.CreateObject("WScript.Network")
    Computer=WShNetwork.ComputerName
end if

Set DDR=CreateObject("SMSResGen.SMSResGen.1")
DDR.DDRNew "System", "CustomAgent", "NES"
DDR.DDRAddString "SMS Unique Identifier", GUID, 64, ADDPROP_NAME
DDR.DDRAddString "Name", Computer, 64, ADDPROP_NAME
DDR.DDRAddString "Netbios Name", Computer, 64, ADDPROP_KEY
Dim IPAddress(3), IPSubnet(3), MACAddress(3)
IPAddress(0)="123.234.12.23"
IPAddress(1)="123.234.12.32"
IPSubnet(0)="123.234.12.0"
IPSubnet(1)="123.234.12.0"
MACAddress(0)="00:02:A5:B1:11:68"
MACAddress(1)="00:02:A5:B1:11:69"
DDR.DDRAddStringArray "IP Addresses", Array(IPAddress(0),IPAddress(1)), 64,
ADDPROP_ARRAY
DDR.DDRAddStringArray "MAC Addresses", Array(MACAddress(0),MACAddress(1)), 64,
ADDPROP_ARRAY
DDR.DDRAddStringArray "IP Subnets", Array(IPSubnet(0),IPSubnet(1)), 64,
ADDPROP_ARRAY
DDR.DDRWrite "MyDDR.DDR"
DDR.DDRSendtoSMS
Set FSO=CreateObject("Scripting.FileSystemObject")
FSO.GetFile("MyDDR.DDR").Delete
```

Creating Status MIF Files

Creating status MIF files in a script can give you more flexibility in the status reporting of complex packages that you distribute with SMS.

To create status MIF files by using scripts, use the MIF Generation Object, which is included with the Active Directory/SMS Synchronization Tools. This tool is available from the Microsoft Web site at http://www.microsoft.com/smserver. You also need ISMIF32.dll, which is available in the %*Windir*% directory of SMS site servers.

Keep in mind the following points when using the MIF Generation Object:

- The script puts the MIF file in the %*Temp*% directory. In the first parameter, do not specify which directory the file should go in.
- ISMIF32.dll and ISMIFCOM.dll must be distributed to the clients where MIF Generation Object-based scripts will be used. You can use SMS software distribution to distribute a batch file that copies ISMIF32.dll to %*Windir*% and then runs the following command:

```
SMSAdsync.exe /s.
```

- To use the MIF Generation Object in a Visual Basic program, add a reference to "ISMIFCOM 1.0 Type Library," and then DIM MIF as a new InstallStatusMIF. As a troubleshooting technique, use ISMIF32.exe, which is available for download at http://www.microsoft.com/smserver, to create a status MIF file by using the same parameters that your script uses. ISMIF32.exe might display a meaningful error message.

Sample C.34 CreateMIF.vbs — creates a status MIF

```
Set MIF=CreateObject("ISMIFCOM.InstallStatusMIF")
Mif.Create "MyScript", "MyCompany", "MyScript", "1.0", " ", " ", "Completed all
phases", 1
```

Scripting Advanced Client Operations

The Advanced Client includes several scripting objects that can be used to control almost every aspect of the Advanced Client. For more information, see the *Microsoft Systems Management Server 2003 Software Development Kit*. Sample C.35 shows some common Advanced Client operations, including:

- Determining the currently assigned site and management point.
- Listing advertisements that are available to the client.
- Listing the state of the client actions and components.
- Starting the hardware inventory collection cycle.
- Listing the client properties.
- Listing the download cache configuration, including downloaded items.
- Changing the download cache size.

Sample C.35 AdvClient.vbs — miscellaneous Advanced Client operations

```
wscript.echo "Client:"
Set smsclient = CreateObject("Microsoft.SMS.Client")
wscript.echo "Assigned to: " & smsclient.GetAssignedSite
wscript.echo "MP: " & smsclient.GetCurrentManagementPoint
wscript.echo ""

set UI = CreateObject("UIResource.UIResourceMgr")
set ads=UI.GetAvailableApplications
wscript.echo ads.count & " available advertisements:"
for i=1 to ads.count
    wscript.echo ads.item(i).packagename
next

wscript.echo ""
wscript.echo "Client actions:"
set mgr = CreateObject("CPApplet.CPAppletMgr")
set actions=mgr.GetClientActions
for each action in actions
    wscript.echo action.name
    if action.name="Hardware Inventory Collection Cycle" then
           action.PerformAction
           wscript.echo "  hardware inventory initiated"
    end if
next

wscript.echo ""
wscript.echo "Client components:"
set components=mgr.GetClientComponents
for each component in components
    if component.state=0 then state="installed"
    if component.state=1 then state="enabled"
    wscript.echo component.displayname, state
next

wscript.echo ""
wscript.echo "Client properties:"
set properties=mgr.GetClientProperties
for each property in properties
    wscript.echo property.name & ": " & property.value
next

wscript.echo ""
set ws=CreateObject("WScript.Shell")
timeoffset=ws.RegRead("HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation
\ActiveTimeBias")
```

(continued)

Sample C.35 AdvClient.vbs — miscellaneous Advanced Client operations *(continued)*

```
set cacheinfo=ui.GetCacheInfo
wscript.echo "Cache Info:"
WScript.echo "Location:          ", cacheinfo.location
WScript.echo "Total Size:        ", cacheinfo.TotalSize
WScript.echo "Free Size:         ", cacheinfo.FreeSize
if cacheinfo.TotalSize=250 then
    cacheinfo.TotalSize=1000 'MB
else
    cacheinfo.TotalSize=250 'MB
end if
wscript.echo "  cache size toggled"
set elements=cacheinfo.GetCacheElements
for each element in elements
    lastreferencetimeadjusted = dateadd( "n", -1 * timeoffset,
element.lastreferencetime  )
    wscript.echo element.cacheelementid, element.contentid, element.contentsize,
element.contentversion, lastreferencetimeadjusted, element.referencecount
    wscript.echo "   ",  element.location
next
```

Debugging Scripts

Debugging scripts is part of the process of script writing. You can often correct errors based on error messages, the behavior of the script, and the context of the problem. However, sometimes you have to examine your script in greater depth as it runs to determine where it is executing in an unexpected manner or where variables are receiving unexpected values. For this level of debugging, a debugging tool is required.

Windows 2000, Windows XP, operating systems in the Windows Server 2003 family, Windows Millennium Edition, and Windows 98 include a scripting debugger. You can also downloaded it from the Microsoft Web site at http://msdn.microsoft.com/scripting/default.htm?/scripting/debugger/dbdown.htm.

► To debug your script by using the script debugger

1. Set the **HKEY_CURRENT_USER\Software\Microsoft\Windows Script\Settings** named value **JITDebug** to **1** (**DWORD**).

   You might have to add both the **Settings** key and the **JITDebug** key.

2. In the debugger, start your script by entering:

   ```
   Cscript.exe //x myscript.vbs
   ```

   Or, to start the debugger and enable the `Stop` statement (in JScript, `Debugger`), enter:

   ```
   Cscript.exe //d myscript.wcs,
   ```

Using Scripts on Web Pages

With only minor modifications, your SMS scripts can be used from Dynamic HTML (DHTML) or ASP Web pages. This allows you to make SMS operations easy to access; they do not have to be restricted to SMS administrators.

Sample C.36 is a DHTML implementation of the part of Sample C.9 that adds the user to a collection. You can create a collection and a package and then advertise the package to the collection. Then users can visit your Web site and select the page that is shown in Sample C.36. In so doing, they add themselves to the collection that you created. SMS then advertises the package to those users, and they receive it like any other SMS advertisement. No intervention is required by an SMS administrator, beyond the initial creation of the collection, package, and advertisement.

Sample C.36 CreateCollection1.html — adds a user to collection

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Sub Document_OnClick

    'customize the next two lines to suit your site and interest
    sitecode="MSO"
    collectionid="SMS00001"

Set Service = Locator.ConnectServer(, "root\SMS\site_" & sitecode)
    'use the following line instead if the user machine is not the site server
(which is usually the case)
'Set Service = Locator.ConnectServer( "servername", "root\SMS\site_" & sitecode,
"username", "password")

'determine the current user, as a resource ID
Set oWshNetwork=CreateObject("Wscript.Network")
    username = oWshNetwork.UserName
Set Users = Service.ExecQuery("Select * From SMS_R_User WHERE Name LIKE ""%" +
username + "%""")
For Each User In Users
    If UCase(User.username) = UCase(username) Then ResID = User.ResourceID
Next
    'you could add an IF statement here in case the user is not discovered yet

Set CollectionRule = Service.Get("SMS_CollectionRuleDirect").SpawnInstance_()
CollectionRule.ResourceClassName = "SMS_R_User"
CollectionRule.RuleName = "ResourceID=" & ResID
```

(continued)

Sample C.36 CreateCollection1.html — adds a user to collection *(continued)*

```
CollectionRule.ResourceID = ResID

    Set oCollection=Service.Get( "SMS_Collection.CollectionID='" & collectionid
& "'" )
oCollection.AddMembershipRule CollectionRule
If Err.Number = 0 Then
               document.all.info.innerText = "You were added to the " &
oCollection.Name & " collection. Any advertisements for this collection will
soon be available to you."
End If

end sub
-->
</SCRIPT>

</HEAD>
<BODY>
<h1>User-Enabled ("Pull") Software Distribution with SMS</h1>

<b><SPAN ID="info">Click on this page to add yourself to the SMS00001
collection.</SPAN></b>
<OBJECT ID="Locator" CLASSID="CLSID:76A64158-CB41-11D1-8B02-00600806D9B6">
</OBJECT>
<OBJECT ID="mysink"  CLASSID="CLSID:75718C9A-F029-11d1-A1AC-00C04FB6C223">
</OBJECT>

</BODY>
</HTML>
```

Understanding Support Implications of Scripted Solutions

Scripts that you create can be supported only by you or people that you work with. Other people, including Microsoft Product Support Services, might voluntarily offer to try to correct problems with your scripts, but you cannot expect that they will provide such support. They might not have a full understanding of the techniques that are used by your script, and your script might be so complex that it would be prohibitively time-consuming to attempt to fix it. For these reasons, scripts should be treated like any other custom in-house application.

Scripts can also cause adverse effects to your systems. Because scripts operate very rapidly, usually without human intervention and without the checks that are built into the SMS Administrator console, your scripts can damage your systems. Thorough analysis of the relevant SMS processes and thorough testing of your scripts minimizes the risk of problems, but it can be difficult to guarantee that no problems will occur. Therefore, you must take responsibility for problems that are caused by your scripts.

In some cases, it might not be clear whether a problem is caused by your script or other factors. You can attempt to reproduce the problem by using normal manual processes. If the problem occurs again, your script is not the cause.

Microsoft provides script, macro, and other code examples for illustration only, without warranty either expressed or implied, including but not limited to the implied warranties of merchantability or fitness for a particular purpose. The scripts provided in this appendix and elsewhere in the SMS documentation are provided as is and Microsoft does not guarantee that the scripts or code can be used in all situations. Microsoft does not support modifications of the scripts or code to suit customer requirements for a particular purpose. While Microsoft support engineers can help explain the functionality of a particular script or code example, they will not modify these examples to provide added functionality nor will they help you construct scripts or code to meet your specific needs. If you have limited programming experience, you might want to consult a Microsoft Solution Provider. Microsoft Solution Providers offer a wide range of fee-based services, including creating custom scripts.

Understanding the Security Implications of Scripted Solutions

Because SMS scripts use the SMS Provider, just like the SMS Administrator console, users running your scripts are subject to the same security constraints that they are subject to when using the SMS Administrator console.

In addition, the SMS Provider generates status messages at the same points that they are created when using the SMS Administrator console. Therefore, SMS operations can be audited to the same degree regardless of whether the operations are done through the SMS Administrator console or by scripts.

Learning More

Additional information about scripting and scripting for SMS is available from a variety of sources:

- Additional information about Visual Basic, Windows Script Host, and VBScript is available in many books.
- An introduction to scripting for beginners is available on the Microsoft Web site at http://www.microsoft.com/downloads/release.asp?ReleaseID=24372.
- Additional information about WSH is available on the Microsoft Web site at http://msdn.microsoft.com/scripting.

The SMS and WMI SDKs are available at the following Web sites:

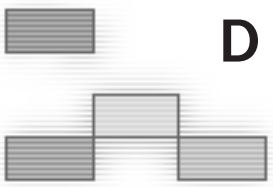
- The *Microsoft Systems Management Server 2003 Software Development Kit* is available on the Microsoft Web site at http://www.microsoft.com/smserver.
- The WMI SDK is available on the Microsoft Web site at http://www.microsoft.com/downloads.

Peer support is available at:

- WMI or SMS newsgroups on msnews.microsoft.com newsgroups.

APPENDIX D

Using SMS in International Organizations



This appendix provides instructions for planning and deploying Microsoft® Systems Management Server (SMS) 2003 in mixed language, international organizations. It is targeted at SMS architects and administrators who are interested in using SMS in a mixed language organization.

For the most current information about international support, see the Release Notes provided on the SMS 2003 product CD. Review any documentation provided with your operating system version concerning multilingual issues.

This appendix does not describe SMS configuration and deployment requirements common to a single-language organization. However, many of the activities you must consider for your multiple-language organization are similar to those described in the following *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide* chapters:

- Chapter 8, "Designing Your SMS Sites and Hierarchy"
- Chapter 10, "Planning Your SMS Deployment and Configuration"
- Chapter 15, "Deploying and Configuring SMS Sites"

In This Appendix

- Planning and Deploying Your Multilingual Site Hierarchy
- Planning and Deploying International Client Packs

Planning and Deploying Your Multilingual Site Hierarchy

SMS provides a broad range of supported languages and other capabilities that enable you to centrally manage a multilingual SMS site hierarchy. For example, SMS supports:

- Multiple language versions of the SMS Administrator console.
- Multiple language versions of the SMS client.
- Double-byte and single-byte character sets for both servers and clients.

The following sections describe how you can use each of these features to plan for and centrally manage a multilingual hierarchy.

This section provides current language support information for each of the seven SMS server languages and lists the client languages supported by each of these packages. You must adequately plan for the language requirements of each site including site server and clients. Each localized site server supports only certain localized clients. After you have fully planned how to address your hierarchies' language needs, you plan, deploy, and configure your sites as recommended in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Planning Multilingual Sites

Many organizations that operate internationally or across linguistic barriers must support multiple languages on their site systems and clients. To address this need, SMS provides several features to help you centrally manage your international SMS site hierarchy and facilitate the flow of information within sites.

Multiple character set support For example, SMS supports both double-byte character sets (DBCS) and single-byte character sets (SBCS) so that you can display and manage both Asian-language and European-language site servers and clients from within the same SMS site hierarchy. DBCS character sets include East Asia languages such as Japanese, Traditional Chinese, Simplified Chinese and Korean. SMS offers seven server user interfaces, four of which are fully localized—that is, fully translated into the target language—and client user interfaces in 22 languages. With SMS, you can centrally manage the desktop computers in your international organization, facilitating the flow of information within organizational units.

Localized interface A localized interface is one that has been adapted to a specific international market or locale. SMS is considered fully localized with respect to the target language when all SMS user interfaces, dialog boxes, and messages to the user are translated into the supported language. SMS supports the display of the language using the correct character set, fonts, and regional settings for a target language.

Enabled interface An enabled interface is one enabled to display and process DBCS/extended characters. An SMS Administrator console displays data from a localized client, but the interface is not translated to the target language. For example, the Korean-language version of SMS is enabled to display and process Korean characters, such as the name of a discovered logon user name. But because it is not localized, the SMS Administrator console is in English, not Korean. The administrator must navigate the menu and dialog text in English. The English-language SMS site server and client versions run on, and are supported by, any localized operating system. The localized versions of SMS site servers and clients are supported only by their matching localized operating systems.

Supported Localized Languages

Table D.1 lists the seven languages that SMS site servers support. The first column lists the languages into which SMS servers are localized or enabled. The second column lists the corresponding client languages supported by each of the server language packages. Any site server language version listed in the table can be the child site or the parent site of any other listed localized server. Except for the French sever, all localized servers support U.S. English clients.

Table D.1 Site Server Languages Supported by SMS

Site server language	Supported client language
English	English and all Windows 2000 Professional and Windows XP Professional languages
French	French
German	German, English
Japanese	Japanese, English
Korean	Korean, English
Simplified Chinese	Simplified Chinese, English
Traditional Chinese	Traditional Chinese, English

The English-language server is available in both English and International English. The contents of the SMS product CD for these two versions are identical and can be used interchangeably; however, their license agreements are different.

Note

The SMS Administrator consoles for Simplified Chinese, Traditional Chinese, and Korean are displayed in the English language, but are enabled for the specific server languages so the DBCS output displays correctly in that language. The SMS client is fully localized to the local language. For example, menus are displayed in English, but data such as package names and inventory data are viewed in the DBCS character set.

In addition to English, SMS clients are available in 21 language versions.

Table D.2 Client Languages Supported by SMS 2003

Czech	Norwegian
Danish	Polish
Dutch	Portuguese
Finnish	Portuguese-Brazilian
French	Russian
German	Simplified Chinese
Greek	Spanish
Hungarian	Swedish
Italian	Traditional Chinese
Japanese	Turkish
Korean	

Character Sets

The DBCS support provided by SMS allows communication to take place between SBCS and DBCS site servers. This functionality is important because it enables you to use a single SMS site hierarchy to centrally manage your SBCS and DBCS sites. For example, you can centrally manage an English-language central site and a Japanese-language child site.

If any sites in your SMS hierarchy will use DBCS:

- SMS server computer names, Microsoft SQL Server™ database names, domain names, Active Directory® site names, SMS site codes, user group names, report names, and SMS accounts that install or manage the SMS site hierarchy cannot contain DBCS characters. Only U.S. English is supported for object names, including those listed, in any SMS hierarchy. Plan to name all network resources with SBCS.
- Hardware inventory class and value names in IDMIF and NOIDMIF files must not contain DBCS. However, values can contain DBCS.
- SMS hardware inventory requires a backslash character (\) to follow every 5c character in a MIF file.
- Software inventory file collection works correctly if 5c characters are in a file name. However, software inventory cannot collect files that contain 5c characters in the file extension.

Folder names or environment variables must not contain extended characters on clients running Microsoft Windows® 98. Extended characters are also called extended ASCII characters.

In addition to providing support for both DBCS and SBCS servers, SMS is available in several language packages that provide different combinations of default server and client component language support. For example, the German-language package of SMS provides a German-language server user interface that supports German-language and English-language clients, while the French-language package of SMS provides a French-language server user interface that supports French-language clients.

Localized versions of SMS site servers are installed only on corresponding localized operating systems. For example, the Japanese language SMS site server version is installed on a Japanese operating system only.

Site Hierarchy Languages

To determine which languages are required at which sites or locations in your organization, you must gather information about:

- Corporate standards for the operating system language version that SMS is installed on.
- Operating system language versions of the clients in each site.
- The language skills of operations personnel at each site location.
- The language skills of end users and associated organizational standards for application localization at each site location.

In each of the SMS sites in your organization, determine which SMS language versions are within your organizational standards. Next, determine what operating system language versions are required by client computers.

- Are there other client operating system language restrictions?
- Who creates and distributes packages?
- Who manages site settings?
- Do your operations personnel require a particular SMS site server language version?

In some situations, you must request exceptions to any corporate standards if the configuration of SMS server and client languages requires language versions outside your corporate standards.

While planning your SMS site hierarchy, plan to name all network resources by using standard ASCII characters, including codes 0 through 127. This includes all computer names, share names, user names, user group names, domain names, and site codes.

After you have fully planned your multilingual SMS site hierarchy and established your sites, administrators and users can effectively use SMS. For example, when an SMS central site administrator in Vancouver uses an English-language version of the SMS Administrator console to send an advertisement to an SMS client in Amsterdam, the Dutch-speaking user reads SMS dialog boxes in her own language because the International Client Pack (ICP) was applied and it provides the localized client software. The administrator of the site sees the Dutch-language computer name and other client data using the correct Dutch-language character display.

Default Collection Names

Any collection, package, or advertisement that is passed down a hierarchy assumes the name it is assigned at the site where it is created, and appears with that same name throughout the hierarchy. The default collections are a special case of this.

As each site is created, the default collections are created at that site, and the name of the collection reflects the language of the site. When the site attaches to a parent site, the collection definitions for the default collections are overwritten with the definitions used at the parent site. If the child and parent use different languages, the collection names are displayed in the language of the parent site. This extends all the way up the hierarchy: default collections are assigned the name used at the central site. The implications of this depend on the code pages used by the affected sites. A code page is a definition of the bit patterns that represent specific letters, numbers, or symbols for character or Unicode data. For additional information about code pages, see the "SQL Server Code Pages" section later in this chapter.

- If the two sites are using the same code page, the SMS Administrator console at the child site displays the collection name correctly, but it is in the language of the central site.
- If the two sites are using different code pages, the SMS Administrator console at the child site cannot display the collection name correctly. The names of the default collections are unreadable.

If the two sites use the same code page, administrators at the child site must learn to recognize the names of the default collections when they are in the language of the central site.

If the two sites use different code pages, at the central site, you can rename the default collections using only ASCII characters. The collection names are then displayed correctly at all sites. In addition, assign ASCII-only names to all collections, packages, and advertisements at any site that is or might later become a parent to a site using a different code page.

An alternative method is to maintain an SMS Administrator console at the child site on a computer that uses the language of the central site.

Site Server Languages

You must plan for language requirements of your site server and its clients concurrently. The language you choose for your SMS site server determines which client localizations it can support. If your organization's policies or operating systems require that the SMS site server and all clients in that site support one language, you can install any non-U.S. English version of SMS that meets your needs. Unlike other localized versions of the SMS server software, the French version only supports its same-language client. All other translated versions of SMS support U.S. English clients. SMS site servers of different installed languages can be placed together in any combination in a hierarchy. If you have a requirement to support multiple client languages within a specific site, then you must install the U.S. English version of the site server and the ICP.

When planning for the support of multiple languages in your SMS site hierarchy, consider the following:

- Which languages are required for clients and servers at which locations in your company.
- When you plan to deploy these languages, keep in mind that some local-language versions of SMS are not released until after the initial U.S. English version is released.
- Which SMS features you need. Balance your requirements against specific language support limitations.
- Whether you will use computer hardware dedicated to SMS site systems or site roles. If you dedicate computer hardware for SMS functions, then you have some flexibility to choose your SMS language version. If you choose not to dedicate computer hardware to SMS site systems or site roles, then you are limited by the language of the operating system of the existing computers.
- Whether the SMS site server is required to run a specific localized operating system.
- Whether you are installing an ICP to support numerous client languages. If you are required to install an ICP at a given site, then you must install the SMS U.S. English version at the site.
- Whether you are able to maintain multiple versions of each language version of SMS. Each language version must have its own service pack and hotfix version. For additional information about SMS versions, see the "Applying updates" section later in this chapter.
- Whether the language skills of the operations staff at each level of the hierarchy are aligned with the requirements of SMS features, such as software distribution, site management, and help desk. Where are packages and advertisements created? In what language? Is the operations staff at lower levels of the hierarchy able to read the names of objects that flow down?

Multilingual Site Hierarchy Example

As an example of a multilingual SMS site hierarchy, consider a company whose headquarters is located in Vancouver, Canada, with branch offices in Tokyo, Seoul, Berlin, and Madrid. The branch offices are all child sites of the parent headquarters site.

The SMS site administrators configure the central, primary site system in Vancouver to use an English-language site server running an English-language SMS Administrator console. In addition, two SMS Administrator consoles are installed on remote workstations running Japanese and Korean operating systems. At the central site in Vancouver, the SMS administrators can view the information of each of the child sites in the native languages and character sets of each of the child sites. The Vancouver site also supports English-language and French-language clients. The ICP is installed at the Vancouver site because the French version of SMS does not support U.S. English clients. All client computers in the site have localized operating systems and users want localized SMS client software, so the ICP is installed to support various client languages.

At the Tokyo site, the SMS administrators install the Japanese version of SMS on a Japanese operating system so they can view data from the Japanese-language clients.

The Seoul site server computer hardware runs a U.S. English operating system. The end users are fluent in U.S. English. Although the client operating systems are Korean, the site administrator decides not to install the Korean version of SMS. The end users use U.S. English clients. The site administrator has access to a Korean version of the SMS Administrator console installed on a Korean operating system that is occasionally used to view localized Korean data.

The Berlin site runs a German-language site server that runs on a German operating system. The site has U.S. English, German, and Traditional Chinese operating system clients. An ICP cannot be installed on the site server because it is not U.S. English. Only German client computers have a localized SMS client. The U.S. English and Traditional Chinese operating system clients use a U.S. English SMS client.

The Madrid site runs an English-language site server that supports a mix of Spanish-language, Italian-language, Turkish-language, and English-language clients. To obtain the required client language support, SMS administrators install ICPs on the Vancouver and Madrid sites only.

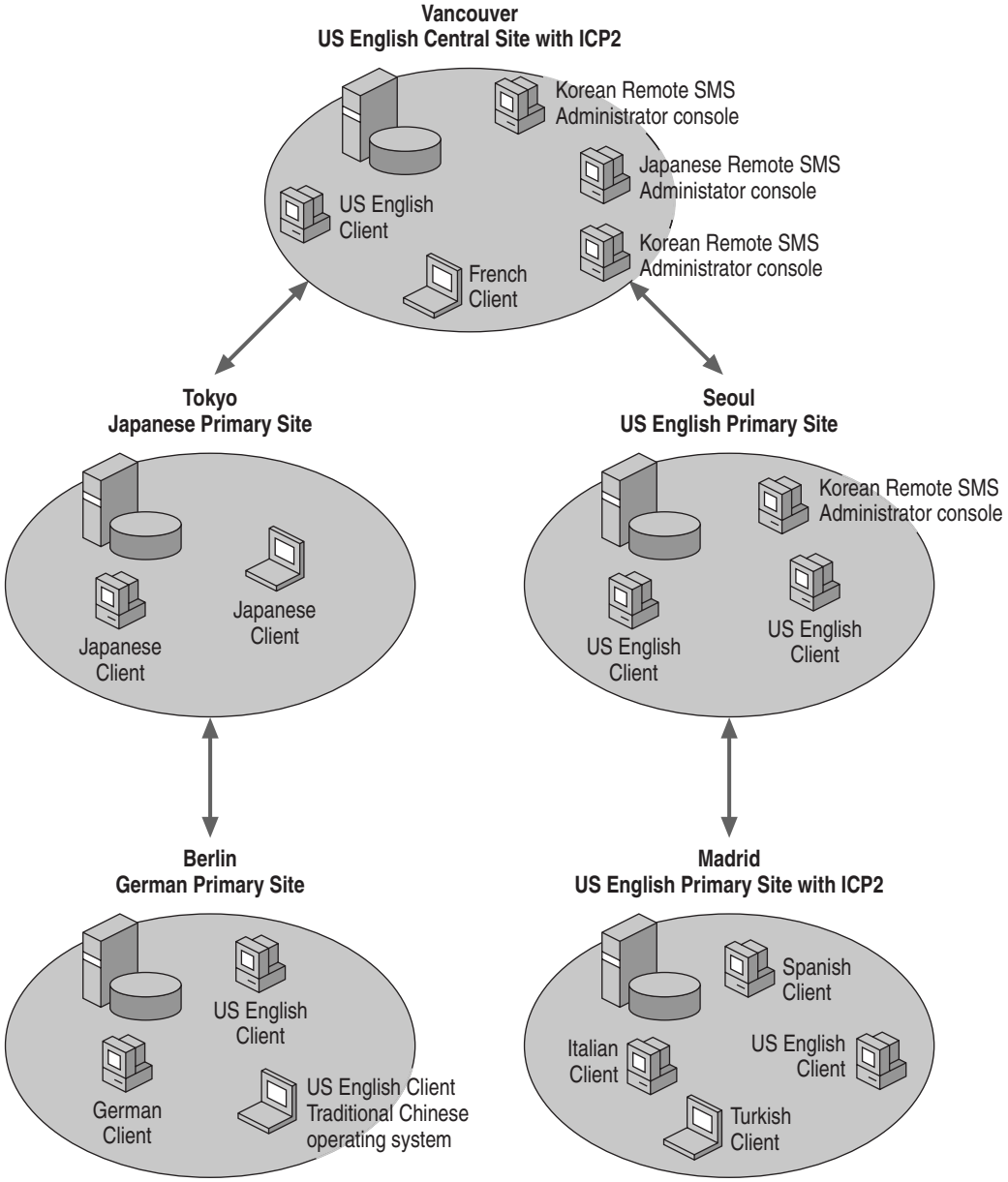
Note

In the case of Korean and Traditional and Simplified Chinese, to view the data correctly, the administrators in the example scenario should install a language-specific operating system, create a site, and then install a language-specific remote SMS Administrator console. The administrators can then view the data of these DBCS languages correctly by using a remote SMS Administrator console to attach to a site server where the data resides. The Japanese site was installed with a localized version of SMS and Japanese operating system, so no remote SMS Administrator console is necessary.

The SMS site administrators of the example company choose this configuration to take advantage of the international features of SMS. Because SMS supports DBCS languages, it supports Japanese (a double-byte character set language) child sites reporting to the English (a single-byte character set language) central site. SMS also supports any DBCS language sites reporting to other DBCS language sites. The European branch has SMS clients enabled with several central, western, and eastern European languages that are deployed by installing ICP2.

For the site administrators, the advantage of this configuration is to have broad administrative coverage of the information systems in an organization. This capability makes it possible for administrators to store all of their SMS information for the organization in one location and to view the data in the native language from each site, from any location, through remote SMS Administrator consoles. Because the site administrators can use any mix of the supported server languages, they have considerable flexibility in designing their SMS site hierarchy to suit the needs of their organization. The English-language site server supports all SMS client languages.

Figure D.1 Example multilingual site hierarchy



Client Languages

SMS attempts to match the language of the SMS client (dialog boxes, messages to the user, and text displayed in any SMS user component) to one of the site server languages supported by SMS. The client operating system can support many language dialects, Australian English for example. If SMS cannot match the client's current language settings directly to a supported language, SMS either renders the client interface as closely as possible in a supported version of the language, or installs client software components in U.S. English, unless the client is French. French SMS site servers support French clients only.

Localization to local-language character, time, and value display takes place at both the operating system level and within SMS. At the operating system level, you can change the regional and language options to specify how the system displays characters and formats numbers, currency, and time and date values in accord with language-specific requirements. Users can set the character mapping of the keyboard to various languages.

International Client Pack

ICPs are installed on U.S. English sites only, and contain the localized client language files required to install local language SMS clients. To install local language SMS Legacy Clients, install an ICP onto your central, primary, or secondary English-language site server, and then install the local language Legacy Clients from that site server. The language-specific client software is copied from the site server to each client access point (CAP), and then onto the clients themselves. Localized SMS client software is installed only on computers running a corresponding localized operating system. The client languages contained within each ICP are listed in Table D.3.

Table D.3 Language Support by ICP Version

International Client Pack	Languages
ICP1	English, French, German, Japanese, and Spanish
ICP2	All the ICP1 languages plus Czech, Danish, Dutch, Finnish, Greek, Hungarian, Italian, Korean, Norwegian, Polish, Portuguese, Portuguese-Brazilian, Russian, Simplified Chinese, Swedish, Traditional Chinese, and Turkish

Client Multilingual User Interfaces

If your SMS sites have an ICP applied and your client computers run Windows 2000 or Windows XP Professional operating systems, SMS offers support for Multilingual User Interface Packs (MUI). The Legacy Client has some requirements that you must address before your users can use MUI. The Advanced Client has none.

Legacy Client

The Legacy Client does not enable MUI support by default. The default SMS client language is English if no match is made with a localized user interface language, except at French sites. French SMS site servers support French clients only. At English sites, the default SMS client language is also English regardless of whether an ICP is installed. There are three cases when the SMS client is set to English on MUI system:

- The client user interface language is English.
- The SMS localized client language files are not on the local client computer.
- The logged on user interface language does not match one of the languages supported by the current System Default Locale, which is the code page referenced in the current System Default Locale.

Requirements To enable MUI support on Legacy Clients running Microsoft Windows 2000 or Microsoft Windows XP Professional, certain requirements must be met.

Each client system must have the **SMSMUIActive** registry entry, the value of which must equal, as follows:

\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SMS\Client\Configuration\Client Properties

SMSMUIActive=REG_DWORD 0x00000001 (1)

The client user must select a user interface language that is supported by the code page referenced in the current System Default Locale, log off, and then log on again so that the SMS client can display the same language as the operating system user interface. For example, a user who has an English Windows 2000 user interface sets the menus and dialogs to German in Regional settings, logs off, and then logs back on. The user then has the German user interface. In this example, the user would have a German SMS client. This level of language switching functionality at logon time is identical to the multilingual user interface language switching that is supported natively in Windows 2000 and Windows XP Professional.

A user account with administrative credentials is required for a user to switch to a language that is not supported by the current System Default Locale. Either the local user or an administrator that is remotely controlling the client can switch languages. After logging on, the local user or the administrator must open the regional settings in Control Panel, set the System Default Locale to the corresponding language, and then restart the system. When the user logs on, the normal SMS localized client installation process starts.

When the user selects a user interface language that uses a code page other than that supported by the system default locale, the logoff and logon level of language switching is not supported. For example, if a user starts an English operating system and then decides to set the menus and dialogs to Japanese in Regional settings, logs off, and then logs on, then the user would have an English version of the SMS client because the system default locale does not support the code page for Japanese.

Advanced Client

The Advanced Client included with each ICP consists of a single Advanced Client installer file, Client.msi. The Client.msi file includes all language resources necessary for clients to switch between supported languages. The default installation language is English. To display an Advanced Client localized installation language, you must install the Advanced Client with the TRANSFORMS command-line option. As an example, you can use the following command from the Japanese language folder:

```
msiexec.exe /i client.msi TRANSFORMS=client.mst
```

For additional information about the TRANSFORMS switch or client language files, see Chapter 4, "Understanding SMS Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*. For additional information about using the Advanced Client Installer, see Chapter 17, "Discovering Resources and Deploying Clients," in the same book.

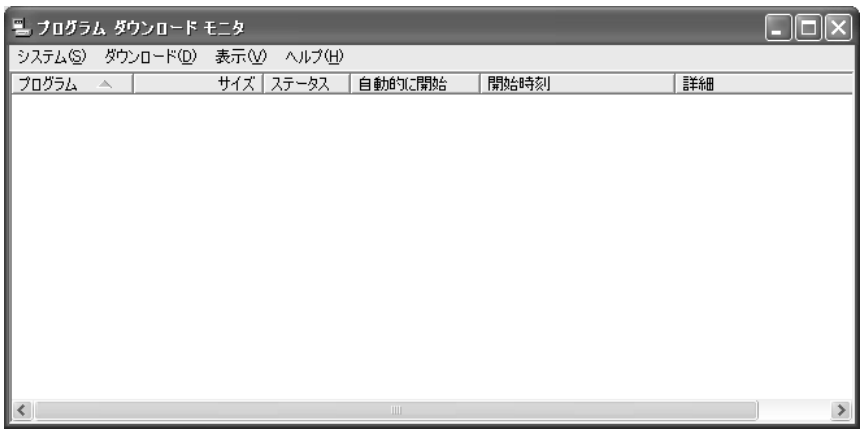
Requirements SMS Advanced Client users or administrators do not need to create a registry key to use MUI. After the Advanced Client is installed, the user can switch from one language to another, within the supported languages, without reinstalling the SMS client software. However, the user must log off the client computer and log on again each time the user changes the display language.

Sample Client User Interfaces

SMS provides your organization great flexibility when planning your site's client composition, considering the large number of different localized clients possible that can report to a single U.S. English site. The following examples show what users on SMS clients with ICPs see as the Advanced Client Control Panel and what they see when a viewing the Program Download Monitor. The examples are Japanese.

Figure D.2 Japanese SMS Advanced Client Control Panel

Figure D.3 Japanese Program Download Monitor



For more information about planning and deploying ICPs, see the "Planning and Deploying International Client Packs" section later in this chapter.

Multilingual Features

SMS 2003 provides international support for all SMS features. However, there are certain issues that you must consider when using SMS Installer and software metering. Planning for the language requirements of your organization might influence the extent to which these features are used.

SMS Installer

You can use SMS Installer to create installation scripts for single and multiple local languages. A local language installation script is a standard SMS installation script that has its entire user interface text, dialog boxes and messages to the user, replaced with the local language equivalents. Table D.4 lists the languages that SMS Installer supports and the corresponding local language installation scripts that it supports. For example, the English-language version of SMS Installer has an English-language user interface, but supports the creation of installation scripts that contain German-language dialog boxes and installation messages. In addition, you can write a multiple language installation script by using the English version of SMS Installer. Such a script might contain installation messages and dialog boxes in English, German, and French.

Table D.4 SMS Installer Local Language Script and Executable Support

SMS Installer user interface language	Supported installation script languages
English	Danish, Dutch, English, Finnish, French, German, Italian, Norwegian, Portuguese-Brazilian, Spanish, Swedish installation messages and dialog boxes
French	French installation messages and dialog boxes

(continued)

Table D.4 SMS Installer Local Language Script and Executable Support *(continued)*

SMS Installer user interface language	Supported installation script languages
German	German and English installation messages and dialog boxes
Korean- enabled (English user interface)	Korean and English installation messages and dialog boxes
Japanese	Japanese and English installation messages and dialog boxes
Simplified Chinese- enabled (English user interface)	Simplified Chinese and English installation messages and dialog boxes
Traditional Chinese- enabled (English user interface)	Traditional Chinese and English installation messages and dialog boxes

Software Metering

Most executable programs contain header information that includes the product version, product name, and language of the program. SMS software metering tracks usage of local-language versions of executable programs by verifying the language field in the version stamp of the executable program file. Verification of the language field is automatically performed by software metering when new software metering rules are created. When you add a software program for monitoring and you change the language field from the default value detected by SMS, you might set the language field incorrectly. If the language field is set incorrectly, the correct language version of the program is not collected. To avoid this problem, verify that the software programs you are metering contain the correct language field entry in accord with your licensing agreements. For information about creating software metering rules, see Chapter 8, "Software Metering." The language version of each monitored program is displayed in the Software Metering Rules Properties page in the SMS Administrator console.

Local Language Display Configuration

Although the English version of the SMS server software supports all the client languages listed in Table D.2, you must perform the following configuration tasks to display non-ASCII characters generated by the clients in the SMS Administrator console.

Locale Setting Must Match the Language of Data

The locale setting of the computer running the SMS Administrator console must match the language of the data you want to view or enter. For example, if you want to view Russian-language hardware inventory data or distribute a software package with a Russian name, the SMS Administrator console you use must run on an operating system on which the locale is set to Russian.

► To set the locale

1. In Control Panel, double-click **Regional Settings**.
2. Click the **Regional Settings** tab, and then select the appropriate locale.

To view data collected from or directed to clients that are using a specific language, you must run the appropriate language version of the SMS Administrator console on a computer set to the corresponding locale.

For each client language other than Japanese, Korean, Simplified Chinese, or Traditional Chinese, inventory data is viewed from an SMS Administrator console by changing the locale setting on a computer running a U.S. English operating system. The system locale setting and operating system language of the computer running the SMS Administrator console must match the language of the data you want to view or enter.

Maintain a Computer Running a Localized Operating System

To view data in Japanese, Korean, Simplified Chinese, or Traditional Chinese, you must maintain a computer running that language. On that computer, install the appropriate localized version of the SMS Administrator console. For example, because the English-language server does not support display of DBCS characters, an administrator in Vancouver who wants to view data from Tokyo must set up a Japanese-localized computer running the Japanese version of the SMS Administrator console to view the data in the Japanese language.

Install Additional Languages on the Computer Where the SMS Provider Is Installed

In addition, you might need to install additional languages to the computer where the SMS Provider is installed. The SMS Provider resides on either the site server or on the computer where the SMS site database is installed. The SMS Administrator console accesses the SMS Provider to view data in these languages.

To install additional languages on Windows 2000 or Windows XP Professional, choose the desired language in **Regional and Language Options** in Control Panel.

SQL Server Configuration

The primary site server and the SMS site database can reside on different computers on your network configured with different languages, provided they use the same code page. If your primary site server has DBCS child sites or clients, remember that SQL Server 2000 or later supports both single-byte and double-byte character sets. All characters from the 21 potential client languages can be stored in the SMS site database.

Language Versions

You can use the English-language version of SQL Server 2000 with the default ANSI 1252 code page with any SMS site server language version, or you can pair local-language versions of SQL Server with a matching localized version of SMS. For example, an SMS administrator in Bonn can use a German-language SMS site server with either a German-language or an English-language version of SQL Server.

Mixed-Language Data

If your database contains mixed-language data, you are limited to a single sort order. You select the sort order during SQL Server setup. For more information, see your SQL Server product documentation.

SQL Server Code Pages

All sites within a single-language hierarchy must use the same SQL Server code page. This is because default collections are unreadable at child sites if the child sites use different code pages than their parent site.

All sites within a multilingual hierarchy must use either ANSI 1252 or ISO 8859-1 code page because all SMS server and ICP languages can use each. The default installation of SQL Server enables the **ANSI to OEM conversion** option. However, because your computer running SQL Server will process data from non-U.S. code pages, you must disable this option. During SQL Server setup, clear the **ANSI to OEM conversion** check box.

Deploying Multilingual Sites

Multilingual sites are installed and upgraded in the same manner as U.S. English sites. The only significant difference between installing the two types of sites is that additional planning is required when you consider creating a mixed-language hierarchy. After you have fully planned how to address your hierarchies' language needs, you can plan, deploy, and configure your sites in the same manner recommended in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

U.S. English sites with ICPs applied differ somewhat from localized sites because the number of possible client languages are increased. With ICP2, a single U.S. English site server can support 22 localized client languages. For additional information about planning and installing ICPs, see the "Planning and Deploying International Client Packs" section later in this chapter.

Sample Deployment Scenarios

The following sample scenarios represent common SMS configurations in multilingual environments.

U.S. English operating systems without ICPs

In this scenario, an international corporation has all its SMS site servers in numerous countries/regions in U.S. English. All SMS clients run on localized operating system languages. However, SMS ICPs are not applied. All users view the U.S. English SMS client. This decision was made to keep all SMS sites at the exact same version, to help minimize administrative operating costs, to have fewer versions and deployments, and to minimize managing localized SMS updates. This is because SMS updates must be requested for each SMS localized version. This example company provides translated screenshots of the U.S. English client, when necessary.

U.S. English operating systems with ICPs

In this scenario, an international corporation has all its SMS site servers in numerous countries/regions in U.S. English. All SMS clients are installed on computers running localized operating systems. ICPs provide SMS clients with a localized interface.

Existing SMS site servers in an existing mixed-language operating system infrastructure

In this scenario, each server operating system and SQL Server code page that it uses presents planning challenges. The servers that are in use now will be used as SMS site servers. The servers might not run localized operating systems. For example, you might find that the server you have been given for use in Kyoto, Japan runs a U.S. English operating system. Your server in Beijing runs a Traditional Chinese operating system. In this example, you will install the U.S. English SMS server in Kyoto. The server in Beijing will continue to run Traditional Chinese, but you discover that the existing SQL Server that you were planning to use in Beijing uses an OEM code page. Because OEM code pages are not supported in a mixed-language hierarchy, you reinstall SQL Server on that server to use the 1252 code page.

New servers in a mixed operating system language infrastructure

In this scenario, you must determine who will administer your computers, and what your corporate policy is regarding operating system language installation. You must consider the language requirements of those that use the SMS Administrator console. Is the current language localized or enabled? Does it make sense to localize SMS on that basis? Can you maintain separate versions, service packs, and hotfixes for the U.S. English, localized language, and ICP versions of SMS? In what languages must you view your clients? If only one language is required, then you can install a localized SMS site server. However, if you require multiple client languages, then you must use the U.S. English version of SMS and apply an ICP on the site server.

Single language environment

If an SMS site consists of computers running a localized operating system version, such as German, and you want to administer the site in German, and you also want to have all clients display a German client interface, then install the German localized SMS version. In this case, there is no need for an ICP. You can add the site to a hierarchy of other SMS language versions. If you have other clients with localized operating systems in the site, such as Italian and Chinese, then those clients will use the U.S. English SMS client.

Planning and Deploying International Client Packs

An ICP displays the user interface elements of the SMS client in the language of the client's operating system. An SMS user who has a localized SMS client and a localized operating system can use SMS to install software and participate in remote control sessions using the SMS client user interface in their own language. Users can also read dialog boxes and menus of the Manual Client Installation Wizard, SMSman.exe, Advanced Client Installer, and the **Systems Management** icon in Control Panel in their own language. ICPs consist of the following localized files:

- Legacy Client
  - Capinst.exe
  - SMSman.exe
  - A few small supporting files
- Advanced Client
  - Client.msi
  - Client.mst
  - A few small supporting files, including language resource files

Deploying an SMS ICP in a large environment is a project that requires planning, testing, and then deployment. The phases of the project include design, testing, and installation.

There are a variety of issues you must design and test for while planning an SMS ICP deployment, such as:

SMS site server localization ICPs are installed only on U.S. English site servers.

Client operating system localization ICPs are available for many different localized operating systems, but not all localized operating systems are supported.

Effect on both your LANs and your WANs Legacy Clients receive ICP software from networked resources.

Effect on the end user Users might be negatively affected if your network cannot adequately support their needs when clients are receiving ICP software.

Deploying an ICP is similar to deploying a service pack. For example, newer program files replace older program files. Deploying an ICP updates CAPs and Legacy Clients using the same mechanism. However, deploying an ICP is dissimilar to deploying a service pack because it does not do anything to the site server except update the client source files. Deploying an ICP does not update files for server components, make changes to the database schema, or perform other non-client upgrade functions.

Planning ICP Deployments

You must plan for each significant event in the deployment. You must also plan for the sequence of events, staffing, scheduling, and system changes. You must analyze the ICP and your environment to assess the impact of the ICP. The procedure you use to deploy the ICP into production is also planned at this stage, and includes steps for monitoring the success of the deployment and for communicating with interested parties. Develop a test plan specifying what ICP functionality you intend to test and what the expected results are.

The following questions reflect some typical issues that you might face when you are planning to deploy the ICP:

- Where do you need to install an ICP?
- Are there steps you must take before applying an ICP?
- Can you roll back an ICP after it is deployed?
- Which method do you want to use to install an ICP?
- Will the deployment of an ICP put a demanding workload on your network or servers?
- How will an ICP affect users, including those who roam and dial in?
- Are clients who dial into the SMS sites forced to upgrade before they can use the connection?
- How can you ensure that the ICP deployment is proceeding successfully?
- Will the application of the ICP regress any fixes previously applied?
- How do you apply subsequent ICPs, service packs, or hotfixes to sites with an ICP already installed?

These questions are answered during the design and test phases of your ICP deployment. After your issues or concerns are addressed, you will be ready to move to the deployment phase.

When you are planning your ICP deployment, refer to the Release Notes that are provided with the ICP. In addition to the release notes, other documentation about changes introduced by the ICP might become available at the same time as the ICP, which might help you to better understand the ICP. Such documentation is included with the ICP and is in the SMS Web site at http://www.microsoft.com/smserver/default.asp.

ICP Design

At a minimum, your ICP deployment design might include preparatory tasks such as:

- Analyzing the ICP to understand the changes it introduces and any required prerequisites. For more information about analyzing the ICP, see the “ICP Requirements” section later in this chapter.
- Analyzing your environment to determine if the deployment will create problems for your users.

- Scheduling the project, including when each site is upgraded and the roles and responsibilities of staff members who assist you.
- Communicating relevant details about the ICP deployment to all interested parties, such as users of SMS services, management, other SMS administrators, and technical staff.
- Preparing to monitor the deployment to ensure that it is successful. For more information about preparing to monitor the deployment, see the "Planning to Monitor" section later in this chapter.
- Preparing the sites, as described in the "Preparing SMS Sites" section later in this chapter.

You will repeat most of these tasks in each of the project phases, including design, testing, and deployment.

ICP Requirements

ICPs include release notes that must be carefully reviewed because they describe prerequisites for the ICP, and they address known problems that the ICP does not solve. When you read the release notes, consider your answers to the following questions:

- Are there any prerequisites that must be implemented before you apply this ICP? Generally, the only prerequisite for an ICP site server installation is that is must be installed on an English-language operating system. However, other requirements might be listed.
- Does this ICP change SMS enough so that you must notify other people who are affected by this change? The people who need notification might include other SMS administrators, managers, technical staff, people responsible for the files and security of the domain controllers, people who monitor bandwidth usage, people who use the SMS services, and the end users.
- Does deploying this ICP cause additional load on the network, servers, or clients? If so, then planning the deployment of the service pack can minimize this effect.
- Is your environment large enough or complex enough to warrant treating the deployment of this ICP as a formal project?
- Do you have problems with clients that might prevent a successful upgrade? If so, you must resolve these issues before upgrading your clients.
- How large are the client upgrade components? To determine the size of your upgrade, see the "Determine the size of client component upgrades" section later in this chapter.

Checking versions

ICPs are released in numbered versions including ICP1 and ICP2. Each version supports a greater number of client operating system languages, including the operating system languages of the earlier version.

When you install an ICP on a U.S. English site, the ICP version must be compatible with a previously installed SMS service pack version. For example, you must use the SMS SP1 version of ICP2 if U.S. English SMS SP1 is installed on the site server. You cannot install the SP2 version of the ICP on a U.S. English SP1 site.

After an ICP is installed, you must be aware of the timing of subsequent ICP releases. ICPs are released shortly after each service pack. When a new service pack is released you must wait for the subsequent ICP release before installing the service pack. After you have obtained the ICP, install the service pack, and then install the ICP. This ensures that Clicore.exe, SMSman.exe, and the other ICP files are placed on CAPs as quickly as possible. Consider disabling any new client installation processes until CAPs are fully updated. This also prevents Legacy Clients from downloading Clicore.exe and SMSman.exe twice — once for the service pack, and again for the ICP.

If you are concerned about the stability of the site after you install the service pack, you can wait for the service pack installation to stabilize before you install the ICP. However, when you later install the localized Legacy Client files on your CAPs, your network will be affected again. Additionally, your international clients will have an older version of the localized client files until the ICP is installed.

Applying updates

To apply client hotfixes, also called product updates, to an ICP site, you must obtain a hotfixed version of the ICP you have installed. You can identify service pack, ICP, and hotfix numbers from the version number. These minor releases and revisions are the last four digits of the Systems Management Server version number.

You can determine whether an ICP is installed by checking for multiple language folders, such as the 00000409 folder for English and the 00000407 folder for German on the site server. There is a folder for each client language supported by that ICP. The language folders reside in the \SMS\BIN\I386\<*language_id*> folders on the Legacy Client.

The first digit of the fourth part of the version number, such as x in x000, is the service pack release number. For example, 2.50.2485.2000 denotes SP2. If the SMS version number is 2.50.2485.3000 or higher, then the service pack is SP3.

Additionally, ICP1 has a 4 as the third-to-last digit. For example, 2.50.2485.2400 indicates SP2 ICP1, and 2.50.2485.3400 indicates SP3 ICP1. Likewise, ICP2 has a 7 as the third-to-last digit. For example, 2.50.2485.3700 indicates SP3 ICP2.

The last three digits are the hotfix version number, which can range from .0001 to .0299.

If you apply SMS SP2 ICP1 to an SMS SP2 U.S. English site that has had several hotfixes installed after SP2 was installed, and files with the same name are included in ICP1, then ICP1 overwrites the newer files because the files in ICP1 do not contain the bug fixes. If the ICP overwrites new files, whatever problems caused you to apply the hotfixes might reappear. For example, you might have previously applied a hotfix to prevent SMSAPM32 from using the CPU at 100 percent. Later you apply ICP1, which does not contain the hotfix. After ICP1 installation, your site server CPU usage is back to 100 percent. To prevent this from occurring, contact Microsoft Product Support Services and obtain the version of the hotfix that correctly matches the ICP you intend to install before ICP installation. Then, after you install the ICP, immediately install the hotfixes that were released later than the ICP.

Caution

Ensure that you install an ICP and ICP hotfixes that correspond to a previously applied SMS service pack and post service pack hotfixes.

Because ICPs are cumulative and are released as they become available, you might need to plan the deployment of language-specific versions of SMS server and client software when the ICP you need becomes available. ICPs have their own setup program and can be installed sequentially onto a site server without any change to the existing clients. All new clients can install in any of the languages from the most recently installed ICP.

Consider the size of the ICPs in relation to the available bandwidth of your network. Later versions of ICPs are larger because each consecutive version contains the languages of the previous versions. Plan the deployment of any language client pack for light network usage periods to minimize degradation of network performance for your users.

Microsoft often releases service packs and client packs on the Web, so that the latest updates can be made available to customers as quickly as possible. The ICPs released by Microsoft contain the current service pack files, in addition to the ICP files, to reduce the number of required installations. These releases are available at http://www.microsoft.com/smserver/default.asp.

Site Server ICP Installation Phases

Installation of SMS ICP occurs in three phases:

Installing an ICP on the site server Installing an ICP starts with running the ICP setup program on a site server. When the site server is upgraded with the ICP, it automatically upgrades the client source files.

Installing an ICP on the CAP and management point The client files on the CAPs and management points are automatically upgraded.

Installing an ICP on a client The Legacy Clients upgrade within 25 hours of the time the CAPs are upgraded, when the Client Configuration Installation Manager (CCIM) on each client runs its maintenance cycle.

Table D.5 Site Component ICP Installation Methods

SMS site component	Method
Site server on primary or secondary site	Running Setup.exe from the ICP source, or package, then manually installing required language packs to the system where the SMS Provider exists.
Site server on secondary site (alternative method)	Running the **Upgrade Secondary Sites** or **Upgrade Site** option from the SMS Administrator console, depending on what kind of site node is selected.
CAP client files on Windows servers	Inbox Manager, which runs automatically.
Distribution points	Not applicable.
Component servers, such as sender servers	Not applicable.

(continued)

Table D.5 Site Component ICP Installation Methods *(continued)*

SMS site component	Method
SQL Server if it is installed on a computer other than the site server, or the SMS Provider if it is installed on the SQL Server	Manually installing the required language packs to the system where the SMS Provider exists. For additional information about manually installing language packs, see the ICP release notes.
SMS Administrator consoles on separate computers	Manually adding code pages.
Legacy Clients	Client Component Installation Manager automatically initiates client update on its 25-hour maintenance cycle, or within an hour of system startup if the files are available on the CAP.

Primary site server ICP installation

The ICP setup program runs on the site server. It stops all components on the site system, copies the client files to the site server, and then restarts all site components on the site system. You can monitor the progress of the setup program with the progress dialog boxes that appear on the site server screen, in the site component manager log, and in the SMSsetup.log, which is in the root of the system drive.

If you use an advertisement to distribute the ICP setup program, ensure that you monitor the status of the advertisement until the ICP installation is complete.

The ICP does not install if Terminal Services is running in execute mode on the site server. The following error message appears when Terminal Services is running in execute mode on the site server:

"Systems Management Server Setup cannot continue because of the following error: Setup has detected that Windows Terminal Server is running on this computer and the current user session is in the execute mode. You must set the current user session to the install mode in order to run SMS Setup."

For Setup to proceed in this case, you must change the user mode to install mode at the command prompt by typing **change user /install**.

Note

If you currently use ICP2 at an existing SMS 2.0 SP4 site and you want to upgrade the site to SMS 2003, you must upgrade to the U.S. English version of SMS 2003 and then apply SMS 2003 ICP2. You do not need to apply ICP1 before you apply ICP2.

Secondary site server ICP installation

You can upgrade secondary sites by using the Upgrade Secondary Site Wizard. The Upgrade Secondary Site Wizard is used to upgrade multiple sites at the same time. You can initiate the Upgrade Secondary Site Wizard from the SMS Administrator console by selecting the **Upgrade Secondary Sites** option from the **Action** menu when a site is selected. The Upgrade Secondary Site Wizard can transfer the necessary files, in an approximately 63-MB package for ICP1, to each site over your WAN. Transferring the files over the WAN can take a long time, depending on the link capacity that is available, so you might want to make the SMS software available at the site on CD.

If you have many sites to upgrade with the ICP, then you might want to consider automating their upgrades. This is described in the "Automating the installation" section later in this chapter.

The CAP upgrade process

All CAPs on the site server are upgraded when you install the ICP. When you install an ICP to a site, it typically takes less than 15 minutes to copy the files to the relevant site directories, restart the services, and copy the updated files to a CAP and logon point over a 100 MB. Table D.6 shows the network traffic required to apply ICPs to a CAP in various scenarios.

Table D.6 Network Traffic Required for CAP Installation

Scenario	Network traffic
New U.S. English CAP installation, no ICP	42 MB
ICP1 update of existing CAP	35 MB
New ICP1 CAP	48 MB
ICP2 update of ICP1 CAP	72 MB

Client ICP installation

The SMS Legacy Client ICP installation process involves installing the base components and the optional components. The base components are upgraded during the first occurrence of the following conditions:

- System startup
- CCIM cycle, which occurs every 25 hours

When the base components are upgraded, Clicore.exe is copied from the site server, runs, and is then recorded in the Ccim32.log and the Clicore.log. The optional components are installed within 60 minutes, and the appropriate files are copied from the CAP.

The base components are contained in Clicore.exe and are installed by CCIM at system startup, or during the next CCIM cycle. All clients, regardless of operating system language version, receive the same version of Clicore.exe. However, when Clicore.exe is run by the CCIM cycle, it detects the language version of the operating system. Then Clicore.exe extracts only the files necessary to support that language into the ms\sms\core\bin folder. All clients install the locally detected installed operating system language plus the U.S. English files. If the local operating system version is not supported by the version of Clicore.exe that has been copied, then the U.S. English files are used. This activity is recorded in the Clicore.log on the client.

During the client base component phase of the upgrade, the effect on the client and the network consist primarily of copying Clicore.exe, and the unpacking process. This occurs when the local operating system language version is detected. Files are then installed on the client. This process can take a little over a minute to run. Table D.7 shows the size of Clicore.exe for each of the ICP versions.

Table D.7 Clicore.exe File Size per ICP Version

ICP version	Clicore.exe file size
U.S. English with no ICP	3.3 MB
ICP1	4.3 MB
ICP2	8.4 MB

When a client starts its upgrade with a new ICP version, you can check the client status by clicking **Systems Management** in Control Panel. The **Components** tab of Systems Management indicates that the based components have changed to an install pending state. The versions of the client agents change to a pending state with a status of repair pending.

The base components take several minutes to install. The optional components take up to 60 minutes for full deployment.

Note

After the ICP is installed on the client, it is normal for optional component versions to differ from the base component versions.

As an example of the network traffic required to deploy the ICP updates to a single client, Table D.8 lists the traffic for different SMS ICPs.

Table D.8 Network Traffic for Client ICP Installation

ICP version	Base component installation	Optional components	Advanced Client
ICP1	5.0 MB	6.9 MB	6.9 MB
ICP2	9.4 MB	11.7 MB	9.8 MB

Client computers running Microsoft Windows NT® 4.0 SP6a, Windows 2000, or Windows XP Professional do not need administrative credentials to successfully install the Legacy Client components because the installation process runs in the context of the Client Service account, which has sufficient privileges. Computers running Windows 98 run in the security context of the logged-on user.

When Legacy Clients running localized operating systems run a logon script, the client is automatically installed with localized client files. There is no need to install ICPs on localized clients.

Normally, Advanced Clients are upgraded by using the same methods that are used to install them. However, Client Push Installation does not automatically upgrade existing Advanced Clients.

Planning to Monitor

You must plan to monitor the deployment to ensure that the project is proceeding successfully. Isolate and resolve problems while they are still small. Design and test reporting solutions during the early phases of your project and use them before you start the deployment in a production environment.

Using SMS Version and Upgrade Information

SMS version data is available from several different sources, and you can use it by employing a variety of techniques. You can choose from among the following techniques those that are appropriate to your situation:

- Try to use reporting-based monitoring techniques if you have many computers or sites, or if you want to monitor the deployment closely.
- Collect version numbers as part of hardware inventory collection if you want to enhance your reporting options, or you could check the registry entries they are stored in.
- Check file versions on random computers, if appropriate.

If you develop reports, then create reports that address the following questions:

- Which sites have been upgraded?
- What percentage of site servers has been successfully upgraded?
- What percentage of CAPs has been successfully upgraded?
- What percentage of Legacy Clients has been successfully upgraded?
- Which site servers were not upgraded?
- Which CAPs were not upgraded?
- Which clients were not upgraded?

The classes listed in Table D.9 provide the version data required by these reports. Techniques for producing these reports are described in this section.

To enhance your reporting options, you can collect client component version numbers during hardware inventory collection. For more information, see the "Check the client component versions" section later in this chapter.

► To determine the version of a file

1. Find the file by using Windows Explorer.
2. Right-click the file and choose **Properties**.
3. On the **Version** tab of the **Properties** dialog box, read the **Product Version** property.

For troubleshooting assistance, view the server logs at the sites during your upgrade.

> **Note**
> The ICP setup program records its activities in the SMSSetup.log file. You can find this file at the root of your system drive.

Preparing SMS Sites

Before you deploy an ICP at any site, consider performing a backup. For additional information about backing up your site, see Chapter 15, "Backup and Recovery." Whether you perform a backup at the site or not is determined by:

- The risk that the ICP installation poses to the site. In the early stages of the deployment the risk is higher than in later stages, especially if you do not thoroughly test the ICP deployment.
- The cost involved in restoring a site to a backup, or rebuilding the site, or if the ICP causes irreparable problems. This varies by site and includes the cost of the work itself and of the lack of SMS services. The most costly site to lose is typically your central site, and the least costly site to lose is a small site at the lowest levels of your hierarchy. The cost also relates to your business dependence on SMS at that site.

The only other preparations required for each site are those required by your project plan or those required by the ICP itself.

ICP Testing

Prepare a network and a set of client and server computers that simulate your production environment as closely as possible. Consider including similar domain models, localized versions of SMS, code pages, localized client operating systems, applications, network links, and roaming scenarios. Apply the ICP using the plans and procedures that you created in the design phase. You might consider performing the tests repeatedly in this environment while you refine your procedures and discover and resolve any identified issues.

Select a portion of your production system that is safe to use for testing but typical enough to represent your production system. Ensure that you do not use a domain managed by multiple sites that you do not want to include in the pilot. Ensure that you are in close proximity to test computers so you can easily reconfigure them. Deploy the ICP during this phase using the plan you create in the design phase including any improvements that you identify in the testing phase. If you observe problems that were not identified in the design phase, adjust your procedures and design accordingly.

ICP Deployment Best Practices

Some aspects of SMS ICP deployment can be done in a variety of ways, but generally, each has a preferred approach that provides the best results.

Use similar code pages

You must use the same code page for all your SMS sites. You must ensure that all sites in a hierarchy can process inventory from all sites below them in the hierarchy, so you must ensure that all code page issues are resolved before applying the ICP. This includes ensuring that the operating system language files are properly installed on all site servers and that all SQL Server computers are using either ANSI 1252 or ISO 8859-1 code page. Ensure that the **Automatic ANSI to OEM conversion** check box is cleared on all computers running SQL Server.

Resolve any CAP problems

Occasionally, permissions problems or other issues can affect the availability of CAPs, especially those remote from the site server itself, which prevents the Site Component Manager service, the Inbox Manager Assistant service, or both from properly servicing the CAPs. Ensure that the CAPs are working properly before installing the ICP. Viewing the status system and log for the Inbox Manager Assistant and Site Component Manager can be helpful.

Apply the ICP hotfixes

ICPs have version numbers that are a version higher than the U.S. English release they upgrade. Great care must be taken to keep hotfixes at the correct version. For more information, see the "Applying updates" section earlier in this chapter.

Conduct research

Peer support from your fellow SMS administrators is an excellent source of knowledge and of solutions based on real-world experience. If you hear from other SMS administrators that an ICP can cause problems under some circumstances, research the issue to decide whether you will apply the ICP. Conduct your own testing and be cautious for any problems you have heard about. If you do observe the problem, then you can contact product support to confirm or deny the issue, and to receive any fixes or workarounds that might be required.

Check the client component versions

You can gather SMS hardware inventory to report data about each Legacy Client component during your SMS ICP client deployment. For each component, the current and pending version is reported in addition to the current state, such as **Not Available**, **Installed**, or **Reboot Required**. This is the same information that is reported on the client in the **Systems Management Properties** dialog box, when you click the **Systems Management** icon in Control Panel. SMS gathers this data during hardware inventory, and it can be displayed in the Resource Explorer under the SMS Client State group class.

You can also query WMI for useful information. The class that the client component version data is available from is SMS_G_System_SMS_CLIENT in the root\SMS\site_<*sitecode*> namespace. The useful properties of the class are:

- ResourceID — a numeric value that can be used to relate the instances to instances for the same resource in other SMS classes
- Component — "Remote Control," for example
- Version — "2.50.2485.1400" for a version of SMS 2003 SP1 ICP1. Your version number will vary.
- State — "Installed"

Apply the ICP to a test computer and determine the actual version number from **Systems Management** in Control Panel.

You can directly check the registry entries collected by this technique by using Registry Editor or a similar program.

Determine the size of client component upgrades

To assess the effect of client component upgrades at large sites or over slow links, it is important to determine the size of the client component upgrades. Multiply the size of the upgrade in MB by the number of clients to be upgraded to determine the total network effect. Divide the resulting number by the capacity of the link, in time, to estimate the length of time required for all clients to be upgraded.

You can measure the size of client component upgrades in various terms, including the size of the files transferred, the increase in disk space used, and the increase in memory used. However, usually the most important issue is the effect on the network of deploying the ICP, especially because you might be doing hundreds or thousands of clients in a short period of time.

Using Network Monitor to determine upgrade size If you need more precise information, then the best approach to determining the size of client component upgrades is to measure the network traffic. To access Network Monitor you must first install it from the SMS compact disc. After installation, click **Start**, then click **Microsoft Network Monitor**, and then click **Network Monitor**.

In a test environment, set up a site server and client with the previous version relative to the ICP you want to install. Configure all site server settings to match those used on your production SMS site server. Then, upgrade the site server, and monitor the upgrade until you are certain the client files on the CAP have been upgraded. You can then start a Network Monitor capture session.

► To start a Network Monitor capture session

1. Start Network Monitor.
2. On the **Capture** menu, click **Buffer Settings**.
3. The **Capture Buffer Settings** dialog box appears.
4. In the **Buffer Size (MB)** box, click **16**, and then click **OK**.

5. Configure a capture session to capture network traffic only between your server and the client computer running Windows NT Workstation 4.0 SP6a or with Windows XP Professional.
6. On the **Capture** menu, click **Start.**

When the capture starts, start the client upgrade by using the client installation method that your site's clients will use. Monitor the client upgrade processing, but do not monitor network traffic, until the client is completely upgraded. You can use the **Systems Management** icon in Control Panel to monitor the upgrade process. When it is completed, return to Network Monitor to stop the capture and analyze the network traffic.

► To analyze the network traffic

1. In the right pane of Network Monitor, the total number of frames transmitted and the total number of bytes captured is displayed.
2. In the Session Statistics pane, displayed in the left middle-window pane, the session traffic between the client and the server is displayed. This indicates how much data, in bytes, was sent from the server to the client, and how much information the client sent back to the server.

Deploying ICPs

When you install an ICP on a U.S. English site server, the client source files on the site server are updated, and then those files are automatically replicated to SMS CAPs. Clients access the site systems to download the client source files. Because the ICP has a version number higher than its corresponding U.S. English version, existing clients detect that an upgrade is available and download the ICP version of the client source files. When new clients are installed, they are installed from the ICP client files that are accessed from the CAP.

ICP Installation

Apply the procedures and details that you have developed in other phases to your entire production system. Ideally, this is done using a staged, systematic plan with continual monitoring. Continual monitoring ensures that any missed problems are dealt with before they affect too many systems or users.

Monitoring the ICP Deployment

Monitoring your ICP deployment is important because you must ensure that the deployment proceeds successfully. If problems occur, you can quickly isolate them.

Monitoring ICP deployments is done using a combination of the status subsystem, log files, and relevant classes in the SMS site database, and hardware inventory. Table D.9 lists items in the SMS Administrator console that are updated when each SMS component is upgraded, classes that reflect upgraded SMS components, and files that can be checked for the correct version number.

Table D.9 Monitoring the ICP Deployment

Updated items	Update status indicators
Site version	To determine which version of the ICP is installed on a site server, check the version value of **HKLM\Software\Microsoft\SMS\Setup** in the registry of the site server. Also, the SMSsetup.log on the site server has an entry that displays the installation version of the ICP. Note: the version information displayed in the SMS Administrator console and SMS_Site class Version property do not include ICP version information.
CAPs – upgrade site components SMS Inbox Manager Assistant by Site Component Manager	Success: Site Component Manager Status Message 1019 followed by Message 500 for the SMS Inbox Manager Assistant Failure: Site Component Manager Status Message 1016, where the component is Inbox Manager Assistant.
Management points	Language directories are created and contain language resource files (.mst) files.
CAPs – upgrade CAP client files	Success: monitor the CAP_<*site*> share on each CAP, checking for the ICP version of Clicore.exe on CAP_<*site*>\clicomp.box\base\<platform> folder, and the Inboxmgr.log Failure: Inbox Manager Component Status Message 3600. There will be many per failed CAP.
Client files on CAPs	Inbox Manager log (Inboxmgr.log). File versions in the Clicomp.box folder.
Clients	Create a custom query and include the Client Version property from the System Resource class. Create a custom report and include the SMS_R_System class, ClientVersion property.
Client components	Click **Systems Management** in Control Panel. The version is displayed on the **Components** tab, in the **Version** column on each client. For more information, see the "Check the client component versions" section earlier in this chapter.

While you monitor the ICP deployment, you can answer the following questions:

- Which sites have been upgraded?
- What percentage of site servers have successfully upgraded?
- What percentage of CAPs has successfully upgraded?
- What percentage of clients has successfully upgraded?

- Which site servers were unsuccessful in the upgrade?
- Which CAPs were unsuccessful?
- Which clients were unsuccessful in the upgrade?

You can use the SMS Administrator console to answer these questions, or you can prepare reports to provide these answers for you, as described in the "Planning to Monitor" section earlier in this chapter. From the reports that list unsuccessful server and client upgrades, you can see where you have problems that require further investigation. You can use the status messages at the site for the appropriate SMS components. You might also want to look at the log files for the relevant components, and randomly check the files on upgraded computers. These systems include the site server, CAPs, secondary sites, and clients. The frequency and depth at which you monitor the deployment will be based on several factors:

- How confident you are that the upgrade will succeed. Good testing and pilot testing, and past experience, will help to increase your confidence.
- How sensitive users at the site are to problems.
- How large or complex the site is, and how likely it is to encounter problems.

Installing the ICP

Installing the ICP involves implementing your deployment plan. The installation at each site proceeds automatically after you run the ICP setup program. However, if the installation affects many computers or more than one physical site, then consider whether you need to control the deployment in an automated fashion.

Automating the installation

If you have many sites, you might want to consider automating the site server ICP upgrade. The Icpsetup.exe has an **/icpunattended** switch that you can use to instruct the setup program to proceed without user interaction, though not silently. You can also install an ICP remotely by using an advertisement or by using Remote Control. You can also install an ICP remotely by using Courier Sender.

A package and advertisement can be created to install the ICP on your primary and secondary sites according to a schedule. If you include the ICP source files in your package, you can configure the package and advertisement as a typical software distribution.

When you automate the installation of ICPs, consider the following issues:

- If you copy the ICP files to the site server or use the local CD drive, and use a package with no source files, you must create a batch file that calls **Icpsetup.exe / icpunattended**.
- The ICP does not install on a server running Terminal Services unless you first change the user mode to install mode at the command prompt using **change user /install**. You might want to incorporate the command into a batch file. The server should be set back to its Terminal Service mode after running the ICP installation. However, you might want to do that after the ICP installation has been completed, and you have verified the installation was successful.
- Even if an advertisement runs successfully on the site server, the ICP might not be installed correctly. You must still monitor the site servers, CAPs, and clients.

ICP setup command-line switch support ICP Setup (ICPSetup.exe) supports the **/ icpunattended** command-line switch. This switch allows ICP Setup to run without user intervention and is used to silently install ICPs on remote or local systems. Unattended installation is supported in the following scenarios:

- Install English SMS, and then install ICP from the CD to a primary or secondary site.
- Install ICP silently by using an advertisement/program sent to a primary or secondary site system that has the ICP CD already in that system's disc drive.
- Install ICP by remote controlling the targeted SMS site system, then run ICP Setup from the ICP CD in that system's disc drive.
- Install ICP on a primary or secondary site silently by creating an SMS package containing the ICP CD files, create a program that uses the Icpsetup.exe **/ icpunattended** switch, and then create an appropriate assigned and unattended advertisement.
- Install ICP silently on a remote SMS primary or secondary site system by using a Courier Sender package.

Using software distribution

If you create a package for the ICP, you can reduce the size of the package by including only the \SMSSETUP folder contents.

Note

An incorrect status message is generated when ICPs are deployed with SMS software distribution. When you install an ICP with SMS software distribution, the status returned for that advertisement always states that the installation failed because Icpsetup.exe reports a non-zero status code. To verify whether the ICP installation has been properly installed, you can check the SMSsetup.log for the actual installation status.

Controlling the ICP installation

Installing an SMS ICP can cause considerable network traffic at large sites or in large organizations and might need to be controlled, because the software is copied to a large number of servers and clients. The load on servers can also be heavy. These effects also occur if your network links are slow.

Table D.10 lists methods that you can use to control the deployment of SMS service packs to the SMS components.

Table D.10 Controlling SMS ICP Deployment to Affected SMS Components

Upgraded SMS component	Method of control
Site server (primary or secondary)	Manually, or by a scheduled advertisement
Secondary site server (optional)	Manually, or by a scheduled job
CAP	The timing of the site upgrade

(continued)

Table D.10 Controlling SMS ICP Deployment to Affected SMS Components *(continued)*

Upgraded SMS component	Method of control
Client files on the CAP	The timing of the site upgrade
Client	Usage strategies, as detailed in the "Legacy Client Upgrades" and "Upgrading Large Sites " sections

CAP upgrades Some organizations try to minimize the issues of Legacy Clients that do not have local site servers by installing CAPs and distribution points at each local site instead. There is no built-in method to control the timing of CAP upgrades within a site. SMS does not support creating remote CAPs. Consider upgrading Legacy Clients connected by a WAN to Advanced Clients, or establishing a secondary site.

Legacy Client upgrades SMS Legacy Client upgrades occur whenever the CCIM cycle runs. The CCIM cycle runs whenever the client is rebooted, or when the SMS Client Service is restarted, then it runs every 25 hours thereafter. Client upgrades are usually not noticed by end-users and do not impose a significant load on your computer infrastructure. However, SMS client upgrades might be a concern if the site is large.

Upgrading large sites If your users usually turn their computers off when they leave work, and if your users tend to arrive for work within a short period of time, such as between 8:00 A.M. and 9:00 A.M., then all your Legacy Clients will attempt to upgrade in that period of time. For sites with very good network capacity and sufficient CAPs, the upgrade is not likely to cause problems. For any other sites, you must consider how you might minimize the effect of the client upgrades.

The most effective strategy is to encourage users to leave their computers turned on at night for a few weeks before the SMS ICP upgrade. They can log off the computers, lock their workstations, or use secure screen savers to ensure that their computers cannot be used during the upgrade. Windows 98 clients should not log off because CCIM will not be able to accomplish its cycle without network access.

During the weeks that users are leaving their computers turned on, many users will have to restart their computers because of the installation of new software or for other reasons. This means that many users will start their 25-hour cycles at different points in the day and will go through a different number of 25-hour cycles. This ensures that on the day the ICP becomes available, the client computers will run CCIM at randomly distributed times throughout the day. This spreads the workload evenly.

Also, if the ICP is installed on a Friday evening, and all computers are locked and logged on to the network, and all computers running Windows 98 are logged on with a secure screensaver, then all client computers will be upgraded over the weekend.

Adding additional CAPS to the site might help if the current CAPs are not sufficient for the anticipated workload. You can remove the additional CAPs when most of the clients are upgraded.

Another alternative is to use the Client Upgrade Control tool (Cliupgrade.exe). The Client Upgrade Control tool is included on the SMS 2003 product CD and is found in \SMSSETUP\BIN\I386. You can use the tool to disable the upgrading of all clients at the site, and then use the tool to enable the upgrading of a subset of the clients each day. For additional information about the Client Upgrade Control tool, see Chapter 4, "Understanding SMS Clients," in the *Microsoft Systems Management Server 2003 Concepts, Planning, and Deployment Guide*.

Troubleshooting the ICP Deployment

Troubleshooting the ICP deployment can occur any time during the deployment. Troubleshooting ICP deployments is much the same as troubleshooting any other problem.

Isolate the problem

Monitoring the deployment alerts you to an existing problem, but it might not specify where the problem is occurring. Is it unique to the particular client or server? Can the client or server obtain the files it requires? Is the client or server having problems installing the files?

To isolate the problem, you might have to begin at the point where you see the symptoms and then work your way backward through SMS processing. Or, you might have to start at the point where you know everything is working properly and then follow the processing forward. For additional information about isolating problems, see the "Site Server ICP Installation Phases" section earlier in this chapter. It is also recommended that you compare computers that are working properly with those that are not working properly.

Understand the problem

When you have located where the problem originates, you must try to understand why the problem is occurring. Status messages from the SMS status system and log files might give helpful details. You can also try to recreate the conditions causing the problem and then check the SMS status system for related error messages. It might be helpful to search the Microsoft Knowledge Base for articles. Try using the event ID and "SMS" as your search parameters. Check the Release Notes for known issues.

Correct the cause of the problem

When you isolate the problem and understand it reasonably well, you can correct it. You might have to adjust your procedures, security constraints, or configuration. Normally, you will not delete files, change registry entries, or perform similar steps, because it might not be practical to do the same manual interventions on all the computers that might be affected by this problem.

If it is not possible to fully understand the problem or isolate it, then you might have to guess what the most likely solutions are and then try each of them. Ensure that solutions you devise do not actually make the problem worse, or create new problems.

If you cannot solve the problem by troubleshooting it yourself, then you might have to call someone else to help. This could include other SMS administrators, or product support.

In the worst-case scenario you can restore your backup so that the site is restored to a known good state. However, this is a serious and difficult step that must be accomplished with very detailed instructions that are appropriate for your particular situation. To restore a site that has parent or child sites, use the recovery procedures documented in Chapter 15, "Backup and Recovery."

Address common problems

To plan the deployment of an SMS ICP, you must decide which issues pertain to your environment, and then plan to minimize the effect of those issues. Consider the issues in Table D.11

Table D.11 Potential Problems in Your SMS Environment

Problem	Potential solution
Client version problems might exist in domains that span multiple SMS sites if Legacy Clients travel from site to site.	Consider the order of site upgrades.
Upgrade size might negatively affect Legacy Clients using with slow network links.	Consider disallowing the use of SMS by these clients, or replace Legacy Clients using slow network links with the Advanced Client.
Bugs might be introduced if you have applied hotfixes that are overwritten by the ICP.	Contact product support to receive the latest ICP containing the corresponding hotfixes.
Network congestion on slow network links between CAPs and their site servers.	Consider the timing of the site upgrades.
Customized hardware inventory and custom reports.	Test the reports, and view the hardware inventory. Extended characters might be used and might cause problems with information displayed.
Customized tools or similar programs using the SMS SDK	Ensure your custom programs still work. Test the programs thoroughly in the international environment, ensuring that no bugs are created by dealing with extended characters and double byte characters.

Index

Special Characters

.cmw (configuration maintenance) files 580
.msp (patch) files 553–554
.mst (transform) files 553, 568–570
.ops (profile settings) files 557, 570
.sms (package definition) files 141, 551

A

ACL Reset 533, 537–538
Actions and Actions Mask flags 661–662
Active Directory
  deployment questions 15–17
  object queries 116
adding boundaries 656–657
administrative patches 554, 577–579
administrative rights installation context 556
administrative tasks
  hardware inventory 45–52
  software inventory 52–59
Administrator console
  checking Advertisement Status 170
  checking Package Status 169–170
  Component Configuration 490
  configuring software distribution component 137–138
  dashboards *See* dashboards
  Distribute Software Updates Wizard 194
  monitoring Advertisement Status 488–489
  monitoring Package Status 484–488
  post-installation tasks 40–41
Administrator console *(continued)*
  Remote Assistance *See* Remote Assistance
  reports *See* reports
  scripting console operations 662–664
  security 133
  Site Status 477–484
  software metering *See* software metering
  Status Reporting 490
  System Status *See* System Status
  Terminal Services *See* Terminal Services
  viewing product compliance data 426
Advanced Clients
  See also clients
  Advanced Client Network Access account 136
  Advanced Client policy 48–49
  advertisements 164
  confirming Remote Tools installations 339–340
  disabling site settings using Remote Tools 357
  enabling BITS 130
  excluding from software metering 313–323
  hardware settings using Remote Tools 358
  installing Remote Tools 336–337
  management point installations 32–33
  multilingual user interfaces 686
  new installations 29–32
  persistent notification 196
  running advertised programs 175–179
  scripting operations 667–669
  security settings using Remote Tools 357
  software metering rules 314

Advanced Configuration attribute 280–287
advanced queries 629–631
Advertised Programs Client Agent 126–128
advertisement object type 110
Advertisement Status 488–489
Advertisement status summarizer 252
advertisements
  adding assignments 642–643
  Advanced Clients 164
  advertisement distribution vs. package distribution 187
  assigned program scenarios 163–164
  assignments based on user logon 163
  checking status 170
  configuring for software updates 231
  creating 159–164, 638–641
  creating for software updates 214
  creating programs that do not require user input 188
  creating with assigned programs 161
  customizing settings for software updates 240
  disabling 165
  event-driven assignments 163
  expanding targets 186
  integrity 165–166
  maintaining 167–168
  modifying 641
  naming 188
  package updates 167
  program dependency 163
  recurring assignments 163
  rerunning 165, 183
  retrying assigned programs 164
  running programs *See* running advertised programs
  scheduled assignments 163
  scripts 638–643
  stopping in emergencies 183
  tasks for managing 158
  unlocking 642
  user-initiated 187
  views 414
AfterBackup.bat 510, 522–523
all-sites level of site status 482–484
analyzing product compliance 424–425
AND logical operator 114
Application Files attribute 277–278
AS keyword 407
attribute class element 111
attribute class joins 114–115
attribute element 111
attribute reference criterion type 112
audit messages 468
authorizing software updates
  configuring advertisements 231
  configuring command-line parameters 227
  configuring scheduled installations 237
  configuring Software Updates Installation Agent advanced options 235
  configuring Software Updates Installation Agent settings 228–231
  creating packages 225–231
  customizing settings for software updates 240
  deploying Office updates 231–240
  distributing Office updates to administrative installations 233
  distributing updates to Windows Installer applications 234–235
  enabling dynamic package configurations 238
  evaluating software updates 224
  expediting approval processing using reference computers 236
  expediting delivery of urgent updates 243–244
  overview 220
  planning packages 221–224
  preparing package source folder 221
  prioritizing software updates 224
  specifying new authorization lists 239–240
  testing 225, 241–243
automating ICP installations 706–707

B

Background Intelligent Transfer Service (BITS) 130
backing up
  site servers 439
  sites *See* backing up sites
  WMI CIM Repository 603–604

backing up sites
See also recovering sites
Backup SMS Site Server task *See* Backup SMS Site Server task
central sites 528
Critical Errors 529
Errors 529
monitoring 528–529
overview 503, 508–509
planning 504
secondary sites 527
third-party backup tools 530
Warnings 529
backup control file 510, 515–522
Backup SMS Site Server task
AfterBackup.bat 510, 522–523
archiving backup snapshots 522–523
backing up sites 513–515
backup control file 510, 515–522
configuring 525–526
content of backup snapshot 513
enabling 525–526
overview 508–513
preparing 513
scheduling considerations 523–525
SMSbkup.ctl 510, 515–522
tasks to perform 513–515
verifying success 526–527
best practices
ICP deployments 702–704
MOF extensions 82–83
software distribution 186–188
software metering 328–330
software updates (distribution) 258–260
software updates (distribution points) 265–266
software updates (end-user experience) 261–262
software updates (general) 254–255
software updates (installation) 260–261
software updates (installation advertisements) 265
software updates (inventory) 257–258
software updates (inventory synchronization) 256–257
best practices *(continued)*
software updates (monitoring) 262
software updates (scheduling) 262–266
software updates (setup) 255–256
BITS (Background Intelligent Transfer Service) 130
browsing tools 598–601

C

CAP (client access point)
preparing for software distribution 128–129
resolving problems 702
site server ICP installation phases 698
upgrades 708
capture filters 371
capture triggers 371
central sites
backing up 528
installations 28–29
site control file backups 507
changing *See* modifying
character sets 676, 678
CIM Repository 603–604, 608–610
CIM Studio 598–599
class qualifiers 49–50
classes
client support 5–7
database 107–108
SMS_UserClassPermissionNames 646
SMS_UserClassPermissions 646
SMS_UserInstancePermissionNames 646
SMS_UserInstancePermissions 646
Win32_Patchstate 194
clearing install flags 442
client access point *See* CAP (client access point)
client agents
hardware inventory 45–52
software inventory 53–61
client patches 554, 579
clients
Advanced Clients *See* Advanced Clients
cannot connect to sites 66
classes 5–7

clients *(continued)*
creating DDRs 665
ICP installations 698–700
languages 684
Legacy Clients *See* Legacy Clients
migration questions 13–15
multilingual user interfaces 684–687
Remote Tools *See* Remote Tools
running advertised programs on regular basis 183
scripting operations 664–669
setting advertisement options 127–128
site configuration questions 23–27
software metering *See* software metering
software updates *See* software updates
support 5–7
supported localized languages 678
user context information 66
.cmw (configuration maintenance) files 580
code pages 702
collecting files
configuring file collection 56
disabling MIF collection 47
enabling MIF collection 47
collecting inventory
hardware *See* hardware inventory
Resource Explorer *See* Resource Explorer
software *See* software inventory
collection limiting 101
collections
choosing existing 132
class-specific methods 636
collection limiting 101
collection-to-collection relationships 633
creating 101–104
creating for software updates 214
creating subcollections 102
creation example 634–636
decreasing evaluation frequency 187
deleting 103, 637
deleting resources 106, 637–638
exporting 103–104
hierarchy 99
collections *(continued)*
importing 103–104
management scope 98
membership rules 96, 633
modifying 102
multilingual deployments 680
multiple dependent subcollections 99
naming 188
overview 95–96
preparing for software distribution 131–133
preparing subcollections for software distribution 132
recovering with direct membership rules 542
removing rules 637
resource management overview 104
resource security 100–101
scripts 633–638
singularly dependent subcollections 99
SMS 2003 vs. SMS 2.0 97
subcollections overview 98
updating memberships 97
updating resource list 105
views 413
compiling SMS Installer-generated executable files 305–306
compliance levels 424
compliance types 424
Component Configuration 490
Component Poller 378
Component Status 477–479
COMPUTE SQL keyword 407
Computer Details page 383, 398–399
computer serial numbers 89
configuration maintenance (.cmw) files 580
configuring
file collection 56
hardware inventory rules 48–52
Remote Tools Client Agent on site servers 335
Remote Tools site-wide settings 340–345
software distribution agent 126–128
software distribution component 137–138
software inventory rules 54–56
status filter rules 492–495

configuring *(continued)*
status summarizers 500
status system *See* configuring status system
configuring status system
Component Configuration 490
configuring status summarizers 500
deleting status messages 500–501
overview 489–490
report detail messages on failure 490
status filter rules 490–500
status messages 491–492
Status Reporting 490
tuning status message configuration 491
connecting to WMI 623–624
controlling software inventory on servers 58
counters, performance monitor 436–437
Create Package from Definition Wizard 173
creating
addresses 653–654
advertisements 159–164, 638–641
classes using NOIDMIF files 72–75
collections 101–104
dashboards 415–418
DDRs for clients 665
packages 139–145, 225–231, 643–644
programs 146–154, 643–644
queries 115–119
query statements 119–124
reports *See* creating reports
scripts 275, 620–621
setup script for packages 145
site systems 658
SMS objects 632
software installation packages with SMS Installer *See* creating software installation packages with SMS Installer
software metering rules 314–323
SQL statements 404–405
status MIF files 667
subcollections 102
creating reports
See also reports
creating dashboards 415–418
enabling reporting points 385
new reports 390–392
overview 379–381, 384–385
tools 385
creating software installation packages with SMS Installer
compiling executable files 305–306
configuring Repackage Installation Wizard 290
custom configuration for repackaging scans 291–292
customizing scripts with Script Editor *See* customizing scripts with Script Editor
distributing executable files 305
installing SMS Installer 275–287
preparing reference computers 288–289
Repackage Installation Wizard overview 287–288
running Repackage Installation Wizard 289–290
running Watch Application Wizard 292–293
Script Editor *See* Script Editor
SMS Installer overview 271–272
SMS Installer process 272–274
SMS Installer run mode 304
SMS Installer tasks 274–275
SMS Installer test mode 304
testing executable files 303–306
Watch Application Wizard overview 292
criterion types 112
critical updates *See* software updates
Custom Installation Wizard 557
custom maintenance tasks 443
Custom Maintenance Wizard 580
custom reports 381
customizing installation attributes 276
customizing product compliance data
automatically 429–432
data file structure 429–430
exporting records to data file 431–432
importing records from data file 431
manually 427–428
overview 427

customizing scripts with Script Editor
creating variables 297–302
destination variable 295
Installation Expert *See* Installation Expert
installation script variables 295–302
Items list 297–302
overview 293
predefined variables 296–297
Script Editor menus 295
Script Editor options 294
unattended setup 303
variable reference 295–296
wrapping existing setup using installation script 303

D

daily site maintenance tasks 444
daily site monitoring tasks
checking clients status 445–446
checking operating system event logs 446
checking site database status 444
checking site server status 445
checking site systems status 445
checking SQL Server error logs 446
checking status filter rules 448
checking system folders 447–448
checking system performance 446
overview 444
dashboards
cloning existing 418
creating 417–418
data 417
deleting 419
modifying 418
overview 381, 415
running 416–417
scheduling 417
viewing list of 415–416
database classes 107–108
date and time operators 113
DDR (discovery data record) 665
debugging scripts 669
deleting
aged collected files 441
aged discovery data 441
aged inventory history 440
aged software metering data 441
aged software metering summary data 442
aged status messages 440
collections 103, 637
dashboards 419
packages 146
programs 154
queries 118
reports 392–393
resources 106, 637–638
SMS objects 632
status messages 500–501
delta replication 158
deploying
ICPs (International Client Packs) 692, 704–710
multilingual sites 690–691
Office XP *See* Office XP
site hierarchies 676
deployment components
hierarchy-specific questions *See* hierarchy-specific deployment questions
site-specific questions *See* site-specific deployment questions
deployment process 4–9
deployment scenarios
in-place upgrades *See* in-place upgrades
new installations *See* new installations
side-by-side upgrades *See* side-by-side upgrades
destination variable in installation script 295
detail messages 468
developing scripts 622
DHTML (Dynamic HTML) 670
disabling
advertisements 165
hardware inventory 45
MIF collection 47
software inventory 53
discovery data record (DDR) 665

discovery views 410–411
displaying distribution point status 628
Distribute Software Updates Wizard
configuring advertisements 231
configuring command-line parameters 227
configuring Software Updates Installation Agent settings 228–231
creating packages 225–231
overview 194
planning packages 221–224
preparing package source folder 221
running 226–227
tasks for authorizing and distributing software updates 220
Distribute Software Wizard 173
distributing
Office XP *See* Office XP
packages 155–158
SMS Installer-generated executable files 305
software *See* software distribution
software updates *See* software update distribution
distribution points
displaying status using scripts 628
enabling protected 130
Manage Distribution Points Wizard 155–157
managing groups 130–131
preparing for software distribution 129–130
scripts 644–646
sending packages 644–646
Dynamic HTML (DHTML) 670
dynamic package configuration 197
dynamic views 409

E

editing query statements 119–124
Elevated Rights Deployment Wrapper (Run_once.exe) 556
embedding properties 652
enabled interfaces 677
enabling
hardware inventory 45
MIF collection 47
Remote Tools Client Agent on site servers 335
enabling *(continued)*
software inventory 53
software metering 312–313
error messages 467
event-driven backups 524–525
event-driven maintenance tasks
adjusting test labs 458
changing Active Directory site boundaries 456
changing hardware or software on site servers 457
changing site server configurations 457
changing SQL Server configurations 457
changing TCP/IP subnets 456
changing WAN infrastructure 457
overview 456
Experts 371, 373–375
Export Object Wizard 118–119, 395–396
extending hardware inventory
changing extensions 86
common MOF extensions 87–93
creating classes using NOIDMIF files 72–75
creating extensions 70
customizing MIF extensions with NOIDMIF files 71
customizing with IDMIF files 75
customizing with MOF files 79–83
MIF extensions overview 71
MOF extensions overview 76
overview 69
propagating extensions throughout hierarchy 70
relationship between Hardware Inventory Client Agent and WMI 77–79
removing extensions 86
requirements for IDMIF files 75–76
scripted extensions 84–86

F

FAQs (frequently asked questions) 582–586
FAT (file allocation system) 202
file collection
configuring 56
viewing collected files 61
firewalls 196–197
flow-of-activity messages 467

Frame Viewer in Network Monitor 373
frames 370
frequently asked questions (FAQs) 582–586
full join 115

H

hardware inventory
  administrative tasks overview 45
  client agents 45–52
  clients cannot connect to sites 66
  compared to software inventory 44
  configuring rules 48–52
  delta 52
  disabling 45
  disabling MIF collection 47
  distributing SMS_def.mof 51
  editing SMS_def.mof 49
  enabling 45
  enabling MIF collection 47
  extending *See* extending hardware inventory
  forcing 46–47
  history 60
  overview 43
  Resource Explorer overview 59
  resynchronization 46
  reviewing data 62–65
  rules 77
  scheduling 46–47
  upgrading SMS 51
  upgrading SMS_def.mof 51
  user context information 66
  viewing 59
  viewing history 60
Hardware Inventory Client Agent 77–79
Hierarchy Maintenance tool (PreInst) 534, 538
hierarchy-specific deployment questions
  Active Directory questions 15–17
  client migration questions 13–15
  network questions 17–19
  overview 8–9
  upgrade questions 10–13
holding sites 6
hotfixes
  finding 89–91
  ICP (International Client Pack) 702

I

ICP (International Client Pack)
  applying updates 695–696
  automating installations 706–707
  CAP problems 702
  CAP upgrades 708
  code pages 702
  deploying 704–710
  deployment best practices 702–704
  deployment requirements 694–696
  design 693–701
  hotfixes 702
  ICP Setup (ICPSetup.exe) 707
  installation 704–710
  Legacy Client upgrades 708
  monitoring deployments 704–706
  planning deployments *See* planning ICP deployments
  planning multilingual sites 684–687
  site server installation phases 696–700
  software distribution 707
  testing 701–704
  troubleshooting deployments 709–710
  upgrading large sites 708
  version checking 694–695
ICP Setup (ICPSetup.exe) 707
IDMIF files
  customizing 75
  example 76
  extensions 71
  requirements 75–76
Import Object Wizard 118–119, 396–397
informational messages 468
inner join 115
in-place upgrades
  deployment scenarios described 27
  deployment steps 33–35
  overview 4, 33

in-place upgrades *(continued)*
questions to ask 9
upgrading sites 35–38
Install on Demand 555–556
installation attributes
Advanced Configuration 280–287
Application Files 277–278
customizing 276
Installation Interface 276–277
Runtime Support 278
System Configuration 279–280
User Configuration 278–279
Installation Expert
creating scripts 275
customized functions 293
overview 272
run mode 304
test mode 304
Installation Interface attribute 276–277
installation script
variables 295–302
wrapping existing setup 303
installation wizards 275
installing
ICP (International Client Pack) 704–710
in-place upgrades *See* in-place upgrades
new installations *See* new installations
post-installation considerations 40–41
Remote Tools 336–340
side-by-side upgrades *See* side-by-side upgrades
SMS Installer 275–287
International Client Pack *See* ICP (International Client Pack)
international deployments *See* multilingual deployments
Internet Control Message Protocol echo 378
interpreting system status 471
introduction 3
Invcif.exe (Microsoft Office Update Database) 195
Invcm.exe (Microsoft Office Update Tool) 195
inventory
hardware *See* hardware inventory
Resource Explorer *See* Resource Explorer
software *See* software inventory
inventory data views 411–412
ISU (Windows Installer Step-up Utility) 271

L

laptop computers 87–89
large queries 630
lazy properties 628–629
left outer join 115
Legacy Clients
See also clients
confirming Remote Tools installations 339–340
disabling site settings using Remote Tools 357
hardware settings using Remote Tools 358–359
installing Remote Tools 336
Legacy Client Software Installation account 135
multilingual user interfaces 685
new installations 29–32
running advertised programs 175, 180–182
security settings using Remote Tools 356
software metering rules 314
upgrades 708
limited queries 630
links in reports 382–384
list of values criterion type 112
local language display configuration 688–689
localized interfaces 676
logical operators 113–114
low rights installations 556

M

maintaining
advertisements 167–168
networks *See* maintaining networks
Office XP installations 577–580
packages 167–168
software metering 326–328
systems *See* maintaining systems

maintaining networks
capture filters 371
capture triggers 371
capturing traffic 372–373
capturing traffic on remote computers 376
diagnostic tools on remote computers 375–376
examining captured data 373
Experts 371, 373–375
installing Network Monitor 372
Network Monitor overview 370
Network Monitor requirements 372
Network Trace 377–378
overview 369
starting Network Monitor 373
tasks 369
maintaining systems
See also monitoring systems
attaching one site to another 460
automating tasks 458
creating child sites 460
daily maintenance tasks 444
developing maintenance plans 434
event-driven maintenance tasks 456–458
maintenance groups 459
maintenance operations 459–463
maintenance tasks overview 435, 437
monitoring maintenance 459
moving site databases 462–463
overview 433–434
periodic maintenance tasks 451–454
rebuilding remote site database server computers 461–462
recovery and repair tools 436
resetting sites by running site reset 463
resources 434–437
sharing custom maintenance tasks 459
swapping site server computers 460–461
throughout hierarchy 458–459
weekly site maintenance tasks 448–450
maintenance operations
attaching one site to another 460
creating child sites 460
moving site databases 462–463
maintenance operations *(continued)*
overview 459–460
rebuilding remote site database server computers 461–462
resetting sites by running site reset 463
swapping site server computers 460–461
Manage Distribution Points Wizard 155–157
Managed Object Format *See* MOF (Managed Object Format)
management data
dynamic 81
static 79
Management Information Format *See* MIF (Management Information Format)
management points
new installations 32–33
preparing for software distribution 128–129
management tools 602–603
managing
advertisements *See* advertisements
collections *See* collections
dashboards 415–419
programs 146–154
queries *See* queries
reports *See* managing reports
resources in collections 104–106
sites after recovery 542–543
software inventory names 57–58
software updates *See* software updates
status filter rules 658–662
WMI setup and upgrades 602
managing reports
See also reports
adjusting time-out settings 402
advanced configuration settings 401–404
changing number of reporting points on Run menu 401
changing number of rows returned by report query 402–403
changing number of values returned by clicking Values 403
Computer Details page 398–399
creating new reports 390–392

managing reports *(continued)*
creating prompts 393
data 389–390
deleting reports 392–393
exporting reports 395–396
importing reports 395–397
integrating prompts 395
locating supplemental reports on disabled reporting points 403–404
modifying prompts 394
modifying reports 390–392
overview 385
predefined reports 390
running reports 387–389
scheduling reports 397–398
Status Message Details page 399–400
supplemental reports 400–401
tools 385
viewing list of reports 385–387
MBSA (Microsoft Baseline Security Analyzer) 195
message date and time 468
message severity 467
Microsoft Baseline Security Analyzer (MBSA) 195
Microsoft Office Inventory Tool for Updates 200, 212–214
Microsoft Office Profile Wizard 557
Microsoft Office Update Database (Invcif.exe) 195
Microsoft Office Update Tool (Invcm.exe) 195
Microsoft Windows 2000 *See* Windows 2000
Microsoft Windows 95 6
Microsoft Windows 98 *See* Windows 98
Microsoft Windows Millennium Edition 6
Microsoft Windows NT 4.0 *See* Windows NT 4.0
Microsoft Windows Server 2003 5
Microsoft Windows XP *See* Windows XP
MIF (Management Information Format)
creating 71
creating classes using NOIDMIF files 72–75
creating status files 667
customizing extensions with NOIDMIF files 71
MIF (Management Information Format) *(continued)*
customizing with IDMIF files 75
disabling collection 47
enabling collection 47
extensions overview 71
requirements for IDMIF files 75–76
status 171–172
milestone messages 468
mixed language deployments *See* multilingual deployments
modifying
advertisements 641
collections 102
dashboards 418
packages 145
programs 154
prompts in reports 394
queries 118
reports 390–392
SMS objects 632
SQL statements 404–405
MOF (Managed Object Format)
best practices for extensions 82–83
collecting SQL Server information 92–93
collecting Windows Installer information 91–92
common extensions 87–93
creating 79
customizing files 79–83
extensions for dynamic data 81
extensions for static data 79
extensions overview 76
extensions with namespaces other than root\CIMv2 81
finding computer serial numbers 89
finding hotfix information 89–91
finding laptop computers 87–89
managing WMI 604–606
relationship between Hardware Inventory Client Agent and WMI 77–79
MOF Compiler (Mofcomp.exe) 603

monitoring
backups 528–529
keys 440
networks *See* monitoring networks
software distribution *See* monitoring software distribution
software update distribution *See* monitoring software update distribution
software update processes *See* monitoring software update processes
System Status *See* monitoring with System Status
systems *See* monitoring systems
monitoring networks
capture filters 371
capture triggers 371
capturing traffic 372–373
capturing traffic on remote computers 376
diagnostic tools on remote computers 375–376
examining captured data 373
Experts 371, 373–375
installing Network Monitor 372
Network Monitor overview 370
Network Monitor requirements 372
Network Trace 377–378
overview 369
starting Network Monitor 373
tasks 369
monitoring software distribution
advertised programs 170
overview 168
package distribution 169–170
status MIFs 171–172
status summaries for packages 169
monitoring software update distribution
component names 247
logging 248–249
overview 244–245
reporting 246–247
status messages 247
tools 245–246
monitoring software update processes
auditing for security vulnerabilities 249–250
checking health of components 252
monitoring status of distributions 250–252
overview 249
troubleshooting errors 253
monitoring systems
See also maintaining systems
daily site monitoring tasks 444–448
developing monitoring plans 434
diagnostic tools 436
log files 435
MOM (Microsoft Operations Manager) 437
overview 433–434
performance monitor counters 436–437
periodic site monitoring tasks 454–456
reports 435
resources 434–437
status system 435
System Monitor 436
weekly site monitoring tasks 450
Windows event logs 436
monitoring with System Status
Advertisement Status 488–489
Component Status 477–479
overview 476
Package Status 484–488
Site Status 477–484
Site System Status 480–484
.msp (patch) files 553–554
MSSecure.XML (Security Patch Bulletin Catalog) 195
.mst (transform) files 553, 568–570
MSXML requirements for software updates 201
MUI pack
Office XP 552
planning multilingual sites 684–687
multilingual deployments
automating ICP installations 706–707
CAP upgrades 708
deploying ICPs 692, 704–710
deploying multilingual sites 690–691
deploying site hierarchies 676

multilingual deployments *(continued)*
ICP installation 704–710
ICP Setup (ICPSetup.exe) 707
Legacy Client upgrades 708
monitoring ICP deployments 704–706
overview 675
planning ICP deployments *See* planning ICP deployments
planning multilingual sites *See* planning multilingual sites
planning site hierarchies 676
sample deployment scenarios 690–691
software distribution 707
troubleshooting ICP deployments 709–710
upgrading large sites 708
Multilingual User Interface pack *See* MUI pack
multiple character set support 676

N

namespaces 49–50
network adapters 370–373, 375–376
network capacity
hardware inventory 45
software inventory 53
network diagrams for site systems 377
Network Monitor
capture filters 371
capture triggers 371
capturing traffic 372–373
capturing traffic on remote computers 376
diagnostic tools on remote computers 375–376
examining captured data 373
Experts 371, 373–375
installing 372
overview 370
planning ICP deployments 703–704
requirements 372
starting 373
Network Monitor Driver 370, 375–376
network performance
hardware inventory 48
software inventory 57–58
Network Trace 377–378, 475
networks
capture filters 371
capture triggers 371
capturing traffic 372–373
capturing traffic on remote computers 376
deployment questions 17–19
diagnostic tools on remote computers 375–376
examining captured data 373
Experts 371, 373–375
installing Network Monitor 372
Network Monitor overview 370
Network Monitor requirements 372
Network Trace 377–378
overview of maintaining and monitoring 369
starting Network Monitor 373
tasks for maintaining and monitoring 369
new installations
Advanced and Legacy Client installations 29–32
central site installations 28–29
deployment scenarios described 27
management point installations 32–33
overview 4, 28
questions to ask 9
NOIDMIF files
creating classes 72–75
customizing MIF extensions 71
NOT logical operator 114
NTFS (NT file system) 202
null value criterion type 112
numerical operators 113

O

O_scan.exe or S_scan.exe (scan component) 199
object type element 111
object types 109–110
Office Inventory Tool for Updates 200, 212–214
Office Profile Wizard 557
Office Update Database (Invcif.exe) 195
Office Update Tool (Invcm.exe) 195
Office XP
administrative installation point creation 566–568
administrative installation source issues 558
administrative patching 577–579

Office XP *(continued)*
administrative rights installation context 556
CD installation issues 558
client configurations 560
client patching 579
configuration maintenance (.cmw) files 580
Custom Installation Wizard 557
Custom Maintenance Wizard 580
customizing source files 566
deploying business requirements 559
deploying concepts 550
deploying in organizations overview 558
deploying overview 548–550
deploying procedure summary 566
deploying software updates 231–240
distributing overview 547
distributing updates to administrative installations 233
Elevated Rights Deployment Wrapper (Run_once.exe) 556
enterprise configurations 559
FAQs (frequently asked questions) 582–586
Install on Demand 555–556
installation progress checking 577
low rights installations 556
maintaining installations 577–580
MUI pack 552
Office Profile Wizard 557
Ohotfix.exe 232, 554
operating system requirements 549
package definition (.sms) files 551
patches 554
planning deployments 560–566
planning for clients without administrative credentials 562–563
planning installation options 564
planning strategies for collections and program advertisements 564–565
preparing Office XP source files 566
profile settings (.ops) files 557, 570
public updates 577–579
resilient sources 580–582
resource kit tools 557
Office XP *(continued)*
service pack distribution 579
SFU overview 551
SFU requirements determination 561–562
software distribution object creation 570–576
standard patching 579
transform (.mst) files 553, 568–570
updating installations 577–580
upgrading required prior to installation 563
Windows Installer 552–556
Windows NT 4.0 low rights installations 556
Ohotfix.exe 232, 554
.ops (profile settings) files 557, 570
OR logical operator 114
ORDER BY keyword 407
order of precedence for queries 114
out-of-schedule backups 524–525
overview 3

P

package access accounts 133–135
package definition (.sms) files 141, 551
package object type 110
Package Status 484–488
packages
advertisement distribution vs. package distribution 187
checking status 169–170
compression 140
creating 139–145, 643–644
creating for software updates 225–231
creating setup script 145
creating source directories 140
creating with SMS Installer *See* creating software installation packages with SMS Installer
customizing settings for software updates 240
deleting 146
delta replication 158
details for specific packages in all sites 486
details for specific packages in single site 487
distributing 155–158
distributing to single user or computer 182

packages *(continued)*
dynamic configuration 197
estimating length of transfers 185
importing package definition files 141
importing Windows Installer 142
integrity 165–166
maintaining 167–168
Manage Distribution Points Wizard 155–157
modifying 145
monitoring distribution 169–170
naming 188
periodic updates 167
planning software updates 221–224
preparing package source folder 221
recovering software update management packages 543
removing 168
scripts 643–646
sending to distribution points 644–646
setting properties 142–145
status summaries 169
summary status for all packages in all sites 484
tasks for managing 139
tasks for preparing to distribute 126
updates during advertisements 167
updates during partially completed downloads 167
views 414
patch (.msp) files 553–554
patches
See also software updates
.msp (patch) files 553–554
administrative 554, 577–579
client 554, 579
Office XP 554
Security Patch Bulletin Catalog (MSSecure.XML) 195
standard 554, 579
Win32_Patchstate 194
performance
Remote Tools 367–368
software metering 329
software updates *See* performance considerations for software updates
performance considerations for software updates
instantaneous loading 269
inventory data 266–267
network issues for mobile clients 269
processing load added to client computers 266
scan component bandwidth 267–268
scan component completeness 268
scan tools 269
status message processing 269
performance monitor counters 436–437
periodic site maintenance tasks
backing up account data 451–452
changing accounts and passwords 452
checking network performance 452
overview 451
performing recovery tests in test lab 453–454
reviewing maintenance plans 453
reviewing security plans 453
periodic site monitoring tasks
checking backup snapshots 455–456
checking hardware 455
checking overall health 455
overview 454
persistent notification 196
per-site level of site status 481–482
ping provider 378
Ping Test
overview 333
testing network connectivity for Remote Tools 351
planning
ICP deployments *See* planning ICP deployments
multilingual sites *See* planning multilingual sites
Office XP deployments 560–566
site backups 504
site hierarchies 676
site recoveries 504
planning ICP deployments
applying updates 695–696
best practices 702–704
CAP problems 702
client component versions 702–703

planning ICP deployments *(continued)*
code pages 702
design 693–701
hotfixes 702
Network Monitor 703–704
overview 692–693
planning to monitor 700
preparing SMS sites 701
requirements 694–696
site server installation phases 696–700
size of client component upgrades 703–704
SMS version and upgrade information 700–701
testing 701–704
version checking 694–695
planning multilingual sites
client languages 684
client multilingual user interfaces 684–687
default collection names 680
enabled interfaces 677
ICP (International Client Pack) 684–687
installing additional languages 689
local language display configuration 688–689
localized interfaces 676
multilingual features 687–688
multiple character set support 676
site hierarchy languages 679–680
site server languages 680–683
SMS Installer 687–688
software metering 688
SQL Server configuration 689–690
supported localized languages 677–679
post-installation considerations 40–41
predefined queries 116
predefined reports 381, 390
predefined site maintenance tasks
backing up site servers 439
clearing install flags 442
deleting aged collected files 441
deleting aged discovery data 441
deleting aged inventory history 440
deleting aged software metering data 441
predefined site maintenance tasks *(continued)*
deleting aged software metering summary data 442
deleting aged status messages 440
monitoring keys 440
overview 438
rebuilding indexes 439
running 438
summarizing software metering file usage data 442
summarizing software metering monthly usage data 442
predefined variables in installation script 296–297
PreInst (Hierarchy Maintenance tool) 534, 538
preparing for Backup SMS Site Server task 513
preparing for site recovery 504–508, 538–542
preparing for software distribution
CAPs (client access points) 128–129
collections 131–133
distribution points 129–130
management points 128–129
security 133–136
preparing for software updates
avoiding problems caused by FAT systems 202
client requirements for test environments 203
configuring synchronization host 214–217
creating advertisements, collections, and programs 214
deploying inventory tools 206–220
distributing inventory tools to client computers 219–220
hardware inventory settings for test environments 204
installation details for components 199
installing Office Inventory Tool for Updates 212–214
installing Security Update Inventory Tool 210–212
MSXML requirements 201
overview 198
performing test inventories 217–218
planning inventory tool deployments 206–210
preinstallation requirements for synchronization component 202
preparing production environments 205–206

preparing for software updates *(continued)*
preparing test environments 203–205
reviewing system requirements for components 198–203
software distribution settings for test environments 204–205
system requirements for inventory tools 199
system requirements for Office Inventory Tool for Updates 200
system requirements for Security Update Inventory Tool 200
verifying inventory tool installations 218
preparing test lab for recovery tests 507–508
product compliance
analyzing 424–425
compliance levels 424
compliance types 424
customizing data automatically 429–432
customizing data manually 427–428
customizing data overview 427
data file structure 429–430
exporting records to data file 431–432
importing records from data file 431
overview 423
solutions 425–426
using data files 430
viewing compliance data 426
profile settings (.ops) files 557, 570
Profile Wizard 557
program object type 110
programs
assigned program scenarios for advertisements 163–164
creating 146–154, 643–644
creating advertised programs that do not require user input 188
creating advertisements 161
creating for software updates 214
deleting 154
managing 146–154
modifying 154
monitoring advertised 170
running advertised *See* running advertised programs
promiscuous mode 370–373, 375
prompted value criterion type 112
prompts in reports
creating 393
integrating 395
modifying 394
overview 381–382
property qualifiers 49–51

Q

qualifiers, class and property 49–51
queries
Active Directory object 116
advanced 629–631
attribute class joins 114–115
copying predefined queries to create new queries 117
creating 115–119
creating statements 119–124
criterion types 112
database classes 107–108
deleting 118
editing statements 119–124
example statements 120–124
exporting 118–119
importing 118–119
large 630
limited 630
logical operators 113–114
modifying 118
multiple object types 124
object types 109–110
optional elements 111–115
order of precedence 114
overview 95, 107
predefined 116
relational operators 113
required elements 111
SMS Query Builder 109
software metering 325
specifying resources in Resource Explorer 68

queries *(continued)*
status 631
Status Message Queries 117
viewing attribute data 108
WQL (WMI Query Language) 109, 115
Query Builder 109
query name element 111
query views 414

R

rebuilding indexes 439
recovering sites
See also backing up sites
ACL Reset 533, 537–538
data traffic issues 539–541
designating reference sites 504, 536
determining whether site recovery is necessary 531
documenting information 505
estimating amount of pending data 542
Hierarchy Maintenance tool (PreInst) 534, 538
intrasite data traffic issues 539–540
managing sites after recovery 542–543
mitigating data loss 542
overview 503, 530
planning 504
preparing for site recovery 504–508, 538–542
preparing test lab for recovery tests 507–508
procedure 532–533
recovering collections with direct membership rules 542
recovering software update management packages 543
Recovery Expert tool 505–506, 533–534
recovery scenarios 531–532
reducing traffic loads 541
security issues 541–542
security settings 506
Site Repair Wizard 504, 533–537
site-to-site traffic issues 540–541
supported configurations 531–532
synchronizing objects 535
tools 533–538
Unenforce Software Metering tool 534
Recovery Expert tool 505–506, 533–534
relational operators 113
Remctrl.log 340
Remote Assistance
See also Remote Tools
configuring settings 340
General tab 341
overview 333
Policy tab 342–343
Security tab 341
starting 333–334
Remote Chat
conducting two-way conversations with clients 350
overview 332
remote computers 375–376
Remote Control
controlling clients 348–350
overview 332
Remote Control Agent (Wuser32.exe) 355–356
Remote Execute
overview 332
running programs on remote clients 351
Remote File Transfer
overview 332
transferring files to and from clients 352
Remote Reboot 332
Remote Tools
See also Remote Assistance; Terminal Services
16-color viewing 367
advanced features (list) 354
Advanced tab 344
client connections 346–348
client diagnostics 333
client hardware settings 357–359
client security settings 356–357
client settings changed by users 353–354
conducting two-way conversations with clients 350
configuring site-wide settings 340–345
confirming installations 339–340
controlling clients using Remote Control sessions 348–350
diagnosing client hardware and software problems 350

Remote Tools *(continued)*
enabling Remote Tools Client Agent on site servers 335
establishing connections using Administrator console 346–347
establishing connections using Remote.exe 347–348
General tab 341
installing on clients 336–340
Notification tab 343
overview 331–333
performance 367–368
Ping Test overview 333
Policy tab 342–343
Remote Chat overview 332
remote client support overview 345
Remote Control overview 332
Remote Execute overview 332
Remote File Transfer overview 332
Remote Reboot overview 332
removing administrator group entries after upgrading 367
restarting remote clients 352
running programs on clients using Remote Execute 351
Security tab 341
testing network connectivity using Ping Test 351
transferring files to and from clients using Remote File Transfer 352
troubleshooting tasks 345
video acceleration 359–366
wallpaper suppression 368
Wuser32.exe (Remote Control Agent) 355–356
Remote Tools Client Agent
configuring on site servers 335
configuring site-wide settings 340–345
confirming installations 339–340
enabling on site servers 335
installing on clients 336–340
Remote Tools Diagnostics 333, 350
Remote.exe 347–348
removing *See* deleting
Repackage Installation Wizard
configuring 290
custom configuration for repackaging scans 291–292
overview 287–288
preparing reference computers 288–289
running 289–290
replication of status summaries 475
Report Viewer
See also reports
Computer Details page 383, 398–399
dashboards *See* dashboards
overview 379–380
running reports 387–389
Status Message Details page 383, 399–400
supplemental reports 400–401
viewing list of reports 385–387
reports
adjusting time-out settings 402
advanced configuration settings 401–404
building SQL statements 405–409
changing number of reporting points on Run menu 401
changing number of rows returned by report query 402–403
changing number of values returned by clicking Values 403
Computer Details page 398–399
creating new reports 390–392
creating prompts 393
creating SQL statements 404–405
custom 381
dashboards *See* dashboards
data 389–390
deleting 392–393
enabling reporting points 385
exporting 395–396
importing 395–397
integrating prompts 395
links 382–384

reports *(continued)*
locating supplemental reports on disabled reporting points 403–404
modifying 390–392
modifying prompts 394
modifying SQL statements 404–405
overview 379–381, 384
predefined 381, 390
prompts overview 381–382
running 387–389
scheduling 397–398
scripts 626–628
site component settings 649–650
software metering 324–325
software update compliance 246, 249
software update distribution status 246, 250–252, 253
software update infrastructure health 247, 252, 253
SQL Server views 409–415
SQL statements overview 404
Status Message Details page 399–400
supplemental 381, 400–401
tools for creating and managing 385
types of 381
viewing list of 385–387
views 414
rerunning advertisements 165, 183
resilient sources 580–582
Resource Explorer
launching 475
monitoring software update distribution 252
overview 59
reviewing inventory data 62–65
running from command line 68–69
specifying explicit resources 68
using collections 69
using queries to specify resources 68
viewing attribute data 108
viewing collected files 61
viewing hardware inventory 59
viewing hardware inventory history 60
viewing software inventory 61
resources
deleting 106
management overview 104
security for collections 100–101
updating collection lists 105
retrieving lazy properties 628–629
reviewing inventory data 62–65
right outer join 115
rules
hardware inventory 48–52
software inventory 54–56
Run_once.exe (Elevated Rights Deployment Wrapper) 556
running
advertised programs *See* running advertised programs
installation wizards 275
predefined site maintenance tasks 438
simple scripts 620–621
running advertised programs
Advanced Clients 176–179
either Advanced or Legacy Clients 175
Legacy Clients 180–182
overview 174
regular basis on clients 183
user context with administrator rights 184
without users being notified 185
Runtime Support attribute 278

S

S_scan.exe or O_scan.exe (scan component) 199
scheduling
Backup SMS Site Server task 523–525
hardware inventory 46–47
reports 397–398
software inventory 54
software metering maintenance tasks 326–328
schema information views 412–413
schemas, WMI 595–597

Script Editor
See also scripts
creating variables 297–302
customizing scripts overview 293
destination variable 295
installation script variables 295–302
Items list 297–302
menus 295
options 294
overview 273
predefined variables 296–297
unattended setup 303
user interface overview 294
variable reference 295–296
wrapping existing setup using installation script 303
Yourapp.exe 305
Yourapp.ipf 306
Yourapp.pdf 306
Yourapp.wsm 306
scripted extensions for hardware inventory 84–86
scripts
See also Script Editor
Actions and Actions Mask flags 661–662
adding boundaries 656–657
additional resources 672
adjusting client agent settings 654–656
adjusting component settings 650–651
Advanced Client operations 667–669
advanced queries 629–631
advertisements 638–643
client operations 664–669
collections 633–638
connecting to WMI 623–624
console operations 662–664
creating addresses 653–654
creating DDRs for clients 665
creating simple 620–621
creating site systems 658
creating SMS objects 632
creating status MIF files 667
customizing with Script Editor *See* customizing scripts with Script Editor
scripts *(continued)*
debugging 669
deleting resources 637–638
deleting SMS objects 632
developing 622
displaying distribution point status 628
distribution points 644–646
embedding properties 652
getting SMS objects 624–631
installation 295–303
large queries 630
limited queries 630
managing status filter rules 658–662
modifying SMS objects 632
overview 615–618
packages 643–646
reporting 626–628
reporting site component settings 649–650
retrieving lazy properties 628–629
running simple 620–621
security rights 646–647
site comments for secondary sites 651–652
SMS site settings 647–649
status queries 631
support 671–672
unattended setup 303
Visual Basic 622–623
Web pages 670–671
writing 618–620
WSH (Windows Script Host) 617–618
secondary sites
backing up 527
site comments in scripts 651–652
security
Administrator console 133
Advanced Client Network Access account 136
client settings using Remote Tools 356–357
collections 100–101
Legacy Client Software Installation account 135
package access accounts 133–135
preparing for software distribution 133–136
recovering sites 506, 541–542

security *(continued)*
rights 646–647
software metering 317
views 415
WMI (Windows Management Instrumentation) 604
Security Patch Bulletin Catalog (MSSecure.XML) 195
security patches *See* patches
Security Update Inventory Tool 200, 210–212
service packs 191, 579
SFU (System Files Update)
overview 551
requirements determination 561–562
side-by-side upgrades
deployment scenarios described 27
deployment steps 38–40
overview 5, 38
questions to ask 9
simple value criterion type 112
site hierarchy languages 679–680
site object type 110
site server languages 680–683
Site Status 477–484
Site System Status 480–484
site views 414
site-specific deployment questions
client configuration questions 23–27
overview 8, 19
site configuration questions 19–23
Skpswi.dat 58–59
.sms (package definition) files 141, 551
SMS Administrator console *See* Administrator console
SMS Advanced Clients *See* Advanced Clients
SMS Hardware Inventory Client Agent 77–79
SMS Installer
Advanced Configuration attribute 280–287
Application Files attribute 277–278
compiling executable files 305–306
configuring Repackage Installation Wizard 290
creating executable files 273–274
creating scripts 275
SMS Installer *(continued)*
custom configuration for repackaging scans 291–292
customizing installation attributes 276
customizing scripts with Script Editor *See* customizing scripts with Script Editor
distributing executable files 305
Installation Expert *See* Installation Expert
Installation Interface attribute 276–277
installation overview 275
options 273
overview 271–272
planning multilingual sites 687–688
preparing reference computers 288–289
process 272–274
Repackage Installation Wizard overview 287–288
run mode 304
running Repackage Installation Wizard 289–290
running Watch Application Wizard 292–293
running wizards 275–276
Runtime Support attribute 278
Script Editor *See* Script Editor
software distribution 172
System Configuration attribute 279–280
tasks 274–275
test mode 304
testing executable files 303–306
User Configuration attribute 278–279
Watch Application Wizard overview 292
SMS Installer-generated executable files
compiling 305–306
creating 273–274
distributing 305
overview 271
process for creating 274
testing 303
SMS Legacy Clients *See* Legacy Clients
SMS Query Builder 109
SMS Remote Tools *See* Remote Tools
SMS Site Repair Wizard 504, 533–537
SMS status system *See* status system

SMS_def.mof
Advanced Clients 48
backing up 49
distributing 51
editing 49
Standard Clients 48
upgrading SMS site 51
SMS_UserClassPermissionNames 646
SMS_UserClassPermissions 646
SMS_UserInstancePermissionNames 646
SMS_UserInstancePermissions 646
SMSbkup.ctl 510, 515–522
software distribution
Administrator console security 133
Advanced Client Network Access account 136
Advertised Programs Client Agent 126–128
advertisement distribution vs. package distribution 187
best practices 186–188
choosing existing collections 132
configuring software distribution agent 126–128
configuring software distribution component 137–138
Create Package from Definition Wizard 173
creating advertised programs that do not require user input 188
creating advertisements 159–164
creating packages 139–145
creating programs 146–154
decreasing collection evaluation frequency 187
deleting packages 146
deleting programs 154
disabling 126, 165
Distribute Software Wizard 173
distributing packages 155–158
enabling 126
enabling BITS 130
enabling protected distribution points 130
estimating length of package transfers 185
expanding advertisement targets 186
integrity for packages and advertisements 165–166
Legacy Client Software Installation account 135
software distribution *(continued)*
maintaining packages and advertisements 167–168
managing distribution point groups 130–131
modifying packages 145
modifying programs 154
monitoring advertised programs 170
monitoring overview 168
monitoring package distribution 169–170
multilingual deployments 707
naming for advertisements, collections, and packages 188
Office XP *See* Office XP
overview 125
package access accounts 133–135
phases 187
preparing CAPs 128–129
preparing collections 131–133
preparing distribution points 129–130
preparing management points 128–129
preparing security 133–136
preparing subcollections 132
rerunning advertisements 165, 183
running advertised programs in user context with administrator rights 184
running advertised programs on Advanced Clients 176–179
running advertised programs on either Advanced or Legacy Clients 175
running advertised programs on Legacy Clients 180–182
running advertised programs on regular basis on clients 183
running advertised programs without users being notified 185
setting advertisement options for clients 127–128
single user or computer 182
SMS Installer 172
status MIFs 171–172
status summaries for packages 169
stopping advertisements in emergencies 183
tasks (list) 182
tasks for managing advertisements 158
tasks for managing packages 139
tasks for preparing for package distribution 126

software distribution *(continued)*
tasks for preparing for software distribution 128
testing 186
user-initiated advertisements 187
Windows Installer-based applications 184
Windows Terminal Services 186
wizards 173–174
software inventory
administrative tasks overview 52
client agents 53–61
clients cannot connect to sites 66
compared to hardware inventory 44
configuring file collection 56
configuring rules 54–56
controlling on servers 58
disabling 53
enabling 53
forcing 54
managing names 57–58
network activity 54
network performance 57–58
overview 43
Resource Explorer overview 59
reviewing data 62–65
scheduling 54
Skpswi.dat 58–59
user context information 66
viewing 61
software metering
adding rules 317–318
best practices 328–330
changes 311–312
concurrent vs. distinct users 324
configuring data collection schedules 328
configuring rules 329–330
creating reports 325
creating rules 314–323
data summarization 324
data usage overview 323
Delete Aged Software Metering Data task 326
Delete Aged Software Metering Summary Data task 326
software metering *(continued)*
deleting rules 318
disabling rules 318
distributing programs to be monitored 328
enabling 312–313
enabling rules 318
excluding Advanced Clients 313–323
file usage summary data 324
inventorying programs to be monitored 328
license compliance 309, 330
monitoring executable programs 310–311
monthly usage summary data 324
overview 309–310
performance 329
planning multilingual sites 688
privacy concerns 330
program version issues 329
queries 325
reporting 324–325
rule matching 315–316, 329
rule properties 314–315
rules in multitiered hierarchies 318–321
rules with same name 321–322
sample reports 325
scheduling data flow 316–317
scheduling maintenance tasks 326–328
security settings 317
SMS 2003 vs. SMS 2.0 311–312
Summarize Software Metering File Usage Data task 327
Summarize Software Metering Monthly Usage Data task 328
Summarize Software Metering tasks 327
Terminal Services 322–323
Software Metering Client Agent
See also software metering
enabling 312–313
monitoring programs 311
overview 310
rule matching 315–316
scheduling data flow 316–317
software metering rule object type 110

software update distribution
configuring advertisements 231
configuring command-line parameters 227
configuring scheduled installations 237
configuring Software Updates Installation Agent advanced options 235
configuring Software Updates Installation Agent settings 228–231
creating packages 225–231
customizing settings for software updates 240
deploying Office updates 231–240
distributing Office updates to administrative installations 233
distributing updates to Windows Installer applications 234–235
enabling dynamic package configurations 238
evaluating software updates 224
expediting approval processing using reference computers 236
expediting delivery of urgent updates 243–244
overview 220
planning packages 221–224
preparing package source folder 221
prioritizing software updates 224
specifying new authorization lists 239–240
testing 225, 241–243
software updates
See also patches
advanced management features 196
basic components functionality 194–195
best practices (distribution) 258–260
best practices (distribution points) 265–266
best practices (end-user experience) 261–262
best practices (general) 254–255
best practices (installation) 260–261
best practices (installation advertisements) 265
best practices (inventory) 257–258
best practices (inventory synchronization) 256–257
best practices (monitoring) 262
best practices (scheduling) 262–266
best practices (setup) 255–256
compared to service packs 191
dynamic package configuration 197
software updates *(continued)*
firewall authentication support 196–197
management challenges 191
management guidelines 192–193
management overview 189–190
management tasks overview 197–198
performance *See* performance considerations for software updates
persistent notification 196
reference computer inventory template 197
scheduled installations 197
tasks for authorizing and distributing *See* tasks for authorizing and distributing software updates
tasks for monitoring distributions *See* tasks for monitoring software update distributions
tasks for monitoring processes *See* tasks for monitoring software update processes
tasks for preparing for *See* tasks for preparing for software updates
technology 195
unattended installations 196
varieties 190
Software Updates Installation Agent 228–231, 235
SQL Server
See also SQL statements
advertisement views 414
collecting information 92–93
collection views 413
compared to WMI 597
discovery views 410–411
inventory data views 411–412
package views 414
planning multilingual sites 689–690
query views 414
report views 414
schema information views 412–413
security views 415
site views 414
status views 413–414
views nomenclature overview 410
views overview 409
views setup 409–410

SQL statements
See also SQL Server
building 405–409
converting Coordinated Universal Time (Greenwich Mean Time) to local time 408
creating for reports 404–405
examples 408–409
modifying for reports 404–405
reports overview 404
variables 408
standard patches 554, 579
static views 409
status filter rules 490–500, 658–662
Status Message Details page 383, 399–400
Status Message Queries 117
Status Message Viewer 469–471, 474–475
status messages
characteristics for identifying source 469
date and time 468
defined 466–467
deleting 440, 500–501
overview 466
severity 467
software update distribution 247
status filter rules 491–500
types of 468
viewing 469–471
status MIF files 667
status queries 631
Status Reporting 490
status summarizers
concepts 472–475
configuring 500
counts 472
display intervals 472–473
launching tools 474–475
Network Trace 475
overview 471
replication 475
Resource Explorer 475
states 472
status indicators 473–474
Status Message Viewer 474–475
status summarizers *(continued)*
thresholds 474
Windows Diagnostics 475
status system
audit messages 468
Component Configuration 490
configuring *See* configuring status system
console items (list) 169
detail messages 468
error messages 467
flow-of-activity messages 467
informational messages 468
interpreting system status 471
message date and time 468
message severity 467
milestone messages 468
monitoring with System Status *See* System Status
overview 465
Status Message Viewer 469–471
status messages 466–471
Status Reporting 490
status summarizers *See* status summarizers
troubleshooting with System Status *See* System Status
warning messages 468
Windows Event Log 501–502
Windows Event Viewer 501–502
status views 413–414
subcollections
creating 102
multiple dependent 99
overview 98
preparing for software distribution 132
singularly dependent 99
subselected values criterion type 112
summarizing software metering usage data 442
supplemental reports 381, 400–401
support for scripted solutions 671–672
supported localized languages 677–679
Syncxml.exe (synchronization component) 199
System Configuration attribute 279–280
System Files Update *See* SFU (System Files Update)
system resource object type 110

System Status
See also status system
Advertisement Status 488–489
all-sites level of site status 482–484
Component Status 477–479
details for specific packages in all sites 486
details for specific packages in single site 487
monitoring overview 476
Package Status 484–488
per-site level of site status 481–482
Site Status 477–484
Site System Status 480–484
summary status for all packages in all sites 484
troubleshooting overview 476

T

tasks for authorizing and distributing software updates
configuring advertisements 231
configuring command-line parameters 227
configuring scheduled installations 237
configuring Software Updates Installation Agent advanced options 235
configuring Software Updates Installation Agent settings 228–231
creating packages 225–231
customizing advertisement and package settings 240
deploying Office updates 231–240
distributing Office updates to administrative installations 233
distributing updates to Windows Installer applications 234–235
enabling dynamic package configurations 238
evaluating software updates 224
expediting approval processing using reference computers 236
expediting delivery of urgent updates 243–244
overview 220
planning packages 221–224
preparing package source folder 221
prioritizing software updates 224
specifying new authorization lists 239–240
testing 225, 241–243
tasks for collecting inventory
hardware inventory 45–52
software inventory 52–59
tasks for maintaining systems
See also tasks for monitoring systems
adjusting test labs 458
automated weekly maintenance tasks 449
automating tasks 458
backing up account data 451–452
backing up application, security, and system event logs 450
backing up site servers 439
changing accounts and passwords 452
changing Active Directory site boundaries 456
changing hardware or software on site servers 457
changing site server configurations 457
changing SQL Server configurations 457
changing TCP/IP subnets 456
changing WAN infrastructure 457
checking network performance 452
clearing install flags 442
custom maintenance tasks 443
daily maintenance tasks 444
deleting aged collected files 441
deleting aged discovery data 441
deleting aged inventory history 440
deleting aged software metering data 441
deleting aged software metering summary data 442
deleting aged status messages 440
deleting unnecessary files 449
deleting unnecessary objects 449
event-driven maintenance tasks 456–458
monitoring keys 440
overview 437–438
performing recovery tests in test lab 453–454
periodic maintenance tasks 451–454
predefined maintenance tasks 438–442
rebuilding indexes 439
reviewing maintenance plans 453
reviewing security plans 453
running disk defragmentation tools 449

tasks for maintaining systems *(continued)*
sharing custom maintenance tasks 459
summarizing software metering file usage data 442
summarizing software metering monthly usage data 442
weekly maintenance tasks 448–450
tasks for monitoring software update distributions
component names 247
logging 248–249
overview 244–245
reporting 246–247
status messages 247
tools 245–246
tasks for monitoring software update processes
auditing for security vulnerabilities 249–250
checking health of components 252
monitoring status of distributions 250–252
overview 249
troubleshooting errors 253
tasks for monitoring systems
See also tasks for maintaining systems
checking available disk space 450
checking available site database space 450
checking backup snapshots 455–456
checking clients status 445–446
checking hardware 455
checking operating system event logs 446
checking overall health 455
checking site database status 444
checking site server status 445
checking site systems status 445
checking SQL Server error logs 446
checking status filter rules 448
checking system folders 447–448
checking system performance 446
daily monitoring tasks 444–448
periodic monitoring tasks 454–456
weekly monitoring tasks 450
tasks for preparing for software updates
avoiding problems caused by FAT systems 202
client requirements for test environments 203
configuring synchronization host 214–217
creating advertisements, collections, and programs 214
deploying inventory tools 206–220
distributing inventory tools to client computers 219–220
hardware inventory settings for test environments 204
installation details for components 199
installing Office Inventory Tool for Updates 212–214
installing Security Update Inventory Tool 210–212
MSXML requirements 201
overview 198
performing test inventories 217–218
planning inventory tool deployments 206–210
preinstallation requirements for synchronization component 202
preparing production environments 205–206
preparing test environments 203–205
reviewing system requirements for components 198–203
software distribution settings for test environments 204–205
system requirements for inventory tools 199
system requirements for Office Inventory Tool for Updates 200
system requirements for Security Update Inventory Tool 200
verifying inventory tool installations 218
tasks for SMS Installer 274–275
temporary capture file 370–371
Terminal Services
See also Remote Tools
overview 333
software distribution 186
software metering 322–323
starting 333–334
testing SMS Installer-generated executable files 303–306
third-party backup tools 530

tools
See also wizards
ACL Reset 533, 537–538
browsing 598–601
CIM Studio 598–599
diagnostic 436
Hierarchy Maintenance (PreInst) 534, 538
Installation Expert *See* Installation Expert
MBSA (Microsoft Baseline Security Analyzer) 195
MOF Compiler (Mofcomp.exe) 603
monitoring software update distributions 245–246
Network Monitor *See* Network Monitor
Network Trace 377–378
Office Inventory Tool for Updates 200, 212–214
Office resource kit 557
Office Update (Invcm.exe) 195
Office Update Database (Invcif.exe) 195
performance monitor counters 436–437
recovery and repair 436, 533–538
Recovery Expert 505–506, 533–534
Remote Tools *See* Remote Tools
reports 385
Script Editor *See* Script Editor
Security Patch Bulletin Catalog (MSSecure.XML) 195
Security Update Inventory Tool 200, 210–212
SMS Installer *See* SMS Installer
software update inventory 199, 206–220
third-party backup 530
Unenforce Software Metering 534
Visual Studio .NET 600
WBEMTest.exe 599
Windows event logs 436
Windows Installer *See* Windows Installer
Windows Installer Step-up Utility (ISU) 271
Winmgmt.exe 603
WMI Command-line (WMIC) 600–601
WMI Control 602
WMI management 602–603
transform (.mst) files 553, 568–570
troubleshooting
Advertisement Status 488–489
Component Status 477–479
ICP deployments 709–710
Package Status 484–488
Remote Tools 345
Site Status 477–484
Site System Status 480–484
software update installation errors 253
System Status overview 476
WMI (Windows Management Instrumentation) 606–612

U

unattended software update installations 196
Unenforce Software Metering tool 534
unspecified object types 110
update rollups *See* software updates
updates *See* software updates
upgrading
in-place SMS upgrades *See* in-place upgrades
Office XP 563
post-installation considerations for SMS 40–41
side-by-side SMS upgrades *See* side-by-side upgrades
WMI (Windows Management Instrumentation) 602
User Configuration attribute 278–279
user group resource object type 110
user resource object type 110
utilities *See* tools

V

v_GS_System inventory data view 411
v_GS_Workstation inventory data view 411–412
variable reference in installation script 295–296
video acceleration
clients running Windows 2000 361
clients running Windows NT 4.0 362–366
enabling 367
overview 359
video compression 360
video compression 360

viewing
attribute data in queries 108
collected files 61
hardware inventory 59
hardware inventory history 60
list of dashboards 415–416
list of reports 385–387
product compliance data 426
software inventory 61
status messages 469–471
Visual Basic scripting 622–623
Visual Studio .NET 600

W

warning messages 468
Watch Application Wizard 292–293
Wbemperm.exe 602
WBEMTest.exe 599
weekly site maintenance tasks
automated 449
backing up application, security, and system event logs 450
deleting unnecessary files 449
deleting unnecessary objects 449
overview 448
running disk defragmentation tools 449
weekly site monitoring tasks 450
WHERE keyword 407
Win32_Patchstate 194
Windows 2000
client classes supported 5
installing Remote Tools 337
Office XP deployments 562
video acceleration on clients 361
Windows 95 6
Windows 98
client classes supported 6
diagnosing client problems using Remote Tools Diagnostics 350
installing Remote Tools 339
Office XP deployments 562
Wuser32.exe (Remote Control Agent) 355–356
Windows Diagnostics 333, 350, 475
Windows Event Log 501–502
Windows Event Viewer 501–502
Windows Installer
applications 184
collecting information 91–92
distributing updates to applications 234–235
importing packages 142
Install on Demand 555–556
Office XP 552–556
patch (.msp) files 553–554
transform (.mst) files 553
versions 552
Windows Installer Source List Resolution 234
Windows Installer Step-up Utility (ISU) 271
Windows Management Instrumentation *See* WMI (Windows Management Instrumentation)
Windows Millennium Edition 6
Windows NT 4.0
client classes supported 5–6
diagnosing client problems using Windows Diagnostics 350
Elevated Rights Deployment Wrapper (Run_once.exe) 556
installing Remote Tools 337
low rights installations 556
Office XP deployments 563
Office XP low rights installations 556
preinstallation testing before installing Remote Tools 338–339
third-party client conflicts for Remote Tools 338–339
video acceleration on clients 362–366
video driver compatibility for Remote Tools 338
Wuser32.exe (Remote Control Agent) 355
Windows Script Host (WSH) 617–618
Windows Server 2003 5
Windows Terminal Services *See* Terminal Services
Windows XP
client classes supported 5
Office XP deployments 562
Winmgmt.exe 603
WINS 371, 374

wizards
Create Package from Definition Wizard 173
Custom Installation Wizard 557
Custom Maintenance Wizard 580
Distribute Software Updates Wizard *See* Distribute Software Updates Wizard
Distribute Software Wizard 173
Export Object Wizard 118–119, 395–396
Import Object Wizard 118–119, 396–397
Manage Distribution Points Wizard 155–157
Office Profile Wizard 557
Repackage Installation Wizard *See* Repackage Installation Wizard
SMS Site Repair Wizard 504, 533–537
Watch Application Wizard 292–293
WMI (Windows Management Instrumentation)
additional resources 613
architecture 591–593
backing up data 603–604
browsing tools 598–601
CIM Repository 603–604, 608–610
CIM Studio 598–599
classes 107–108
Command-line tool 600–601
compared to SQL Server 597
connecting to WMI using scripts 623–624
connectivity issues 610
Hardware Inventory Client Agent relationship 77–79
installation issues 610
management tools 602–603
managing 601–606
WMI (Windows Management Instrumentation) *(continued)*
MOF Compiler (Mofcomp.exe) 603
MOF files 604–606
Object Model 593–595
overview 587–591
programming issues 611
resource consumption issues 611
schemas 595–597
security 604
setup 602
troubleshooting 606–612
upgrading 602
Visual Studio .NET 600
Wbemperm.exe 602
WBEMTest.exe 599
Winmgmt.exe 603
WMI Control tool 602
WQL (WMI Query Language) 109, 115
WMI Command-line (WMIC) tool 600–601
WMI Control tool 602
WMI Query Language (WQL) 109, 115
WSH (Windows Script Host) 617–618
Wuser32.exe (Remote Control Agent) 355–356

Y

Yourapp.exe 305
Yourapp.ipf 306
Yourapp.pdf 306
Yourapp.wsm 306