

# Windows Server® 2008 Security Resource Kit

*Jesper M. Johansson and  
MVPs with the Microsoft  
Security Team*

To learn more about this book, visit Microsoft Learning at  
<http://www.microsoft.com/MSPress/books/11841.aspx>

9780735625044

**Microsoft®**  
Press

# Table of Contents

Acknowledgements .....xv

Introduction .....xvii

**Part I    Windows Security Fundamentals**

**1    Subjects, Users, and Other Actors .....3**

    The Subject/Object/Action-Tuple ..... 3

    Types of Security Principals..... 4

        Users..... 4

        Computers..... 7

        Groups ..... 7

        Abstract Concepts (Log-on Groups) ..... 10

        Services ..... 11

    Security Identifiers ..... 12

        SID Components..... 12

        SID Authorities ..... 13

        Service SIDs..... 14

        Well-Known SIDs ..... 15

    Summary ..... 16

    Additional Resources ..... 16

**2    Authenticators and Authentication Protocols..... 17**

    Something You Know, Something You Have ..... 17

        Something You Know ..... 18

        Something You Have ..... 18

        Something You Are ..... 18

    Understanding Authenticator Storage ..... 19

        LM Hash..... 21

        NT Hash ..... 23

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

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

[www.microsoft.com/learning/booksurvey/](http://www.microsoft.com/learning/booksurvey/)

Password Verifier .....	24
In Memory .....	25
Reversibly Encrypted .....	27
Authentication Protocols .....	29
Basic Authentication .....	29
Challenge-Response Protocols .....	30
Smart Card Authentication .....	37
Smart Cards and Passwords .....	38
Attacks on Passwords .....	38
Obtaining Passwords .....	38
Using the Captured Information .....	42
Protecting Your Passwords .....	44
Managing Passwords .....	46
Use Other Authenticators .....	46
Record Passwords, Safely .....	46
Stop Thinking About Words .....	47
Set Password Policies .....	47
Fine-Grained Password Policies .....	49
Summary .....	54
Additional Resources .....	54
<b>3 Objects: The Stuff You Want .....</b>	<b>55</b>
Access Control Terminology .....	55
Securable Objects .....	56
Security Descriptors .....	56
Access Control List .....	58
Access Control List Entry .....	59
Access Masks .....	61
Relationship Between Access Control Structures .....	66
Inheritance .....	66
Security Tokens .....	70
Access Check Process .....	72
Integrity Labels .....	74
Empty and NULL DACLs .....	75
Security Descriptor Definition Language .....	75
Tools to Manage Permissions .....	79
caccls and icaccls .....	79

SC .....	81
subinacl .....	81
Major Access Control Changes in Windows Server 2008.....	81
TrustedInstaller Permissions .....	81
Network Location SIDs .....	82
File System Name Space Changes.....	82
Power User Permissions Removed.....	82
OWNER_RIGHT and Owner Rights .....	82
User Rights and Privileges.....	83
RBAC/AZMAN.....	88
Summary .....	88
Additional Resources .....	89
<b>4     Understanding User Account Control (UAC).....</b>	<b>91</b>
What Is User Account Control? .....	92
How Token Filtering Works.....	92
Components of UAC .....	94
UAC Elevation User Experience .....	94
Application Information Service .....	98
File and Registry Virtualization .....	98
Manifests and Requested Execution Levels .....	100
Installer Detection Technology.....	101
User Interface Privilege Isolation.....	102
Secure Desktop Elevation Prompts .....	102
Using Remote Assistance .....	103
UAC Remote Administrative Restrictions .....	103
Mapping Network Drives When Running in Admin Approval Mode .....	104
Application Elevations Blocked at Logon.....	106
Configuring Pre-Windows Vista Applications for Compatibility with UAC.....	107
UAC Group Policy Settings .....	108
UAC Policy Settings Found Under Security Options.....	108
Related UAC policies .....	110
What's New in UAC in Windows Server 2008 and Windows Vista SP1 .....	111
New Group Policy Setting: UIAccess Applications to Prompt for Elevation without Using the Secure Desktop.....	112
UAC Prompt Reduction When Performing File Operations in Windows Explorer .....	112

	More Than 40 Additional UAC-Related Application Compatibility Shims .....	112
	UAC Best Practices .....	112
	Good Practice .....	112
	Better Practice .....	113
	Best Practice .....	113
	Summary .....	113
	Additional Resources .....	114
<b>5</b>	<b>Firewall and Network Access Protection .....</b>	<b>115</b>
	Windows Filtering Platform .....	116
	Windows Firewall with Advanced Security .....	118
	Improvements in the Windows Firewall .....	118
	Managing the Windows Firewall .....	122
	Routing and Remote Access Services .....	130
	Improvements in RRAS .....	131
	Internet Protocol Security .....	133
	IPsec Basics .....	133
	New Capabilities in Windows Server 2008 .....	136
	Network Access Protection .....	139
	Architecture .....	140
	NAP Implementation .....	143
	NAP Scenarios .....	146
	Summary .....	150
	Additional Resources .....	150
<b>6</b>	<b>Services .....</b>	<b>151</b>
	Introduction to Services .....	151
	What Is a Service? .....	152
	Service Logon Account .....	152
	Service Listener Ports .....	154
	Configuring Services .....	155
	Windows Server 2008 Services by Role .....	161
	Attacks on Services .....	161
	Blaster Worm .....	161
	Common Service Attack Vectors .....	163
	Service Hardening .....	165
	Least Privilege .....	165

Service SIDs . . . . .	170
Write Restricted SIDs . . . . .	172
Restricted Network Access . . . . .	174
Session 0 Isolation . . . . .	176
Mandatory Integrity Levels . . . . .	176
Data Execution Prevention . . . . .	176
Other New SCM Features . . . . .	177
Securing Services . . . . .	178
Inventory Services . . . . .	178
Minimize Running Services . . . . .	178
Apply a Least-Privilege Model to Remaining Services . . . . .	179
Keep Your Updates Up To Date . . . . .	179
Creating and Using Custom Service Accounts . . . . .	180
Use Windows Firewall and IPsec for Network Isolation . . . . .	181
Auditing Service Failures . . . . .	181
Develop and Use Secure Services . . . . .	182
Summary . . . . .	182
Additional Resources . . . . .	182
<b>7 Group Policy . . . . .</b>	<b>183</b>
What Is New in Windows Server 2008 . . . . .	183
Group Policy Basics . . . . .	184
The Local GPO . . . . .	184
Active Directory-Based GPOs . . . . .	185
Group Policy Processing . . . . .	190
What Is New in Group Policy . . . . .	194
Group Policy Service . . . . .	194
ADMX Templates and the Central Store . . . . .	194
Starter GPOs . . . . .	197
GPO Comments . . . . .	198
Filtering Improvements . . . . .	199
New Security Policy Management Support . . . . .	201
Windows Firewall with Advanced Security . . . . .	204
Wired and Wireless Network Policy . . . . .	206
Managing Security Settings . . . . .	208
Summary . . . . .	212
Additional Resources . . . . .	212

<b>8</b>	<b>Auditing. . . . .</b>	<b>213</b>
	Why Audit? . . . . .	213
	How Windows Auditing Works . . . . .	214
	Setting an Audit Policy . . . . .	216
	Audit Policy Options . . . . .	221
	Developing a Good Audit Policy . . . . .	224
	New Events in Windows Server 2008 . . . . .	226
	Using the Built-In Tools to Analyze Events . . . . .	230
	Event Viewer . . . . .	231
	WEvtUtil.exe . . . . .	236
	Summary . . . . .	237

## **Part II   Implementing Identity and Access (IDA) Control Using Active Directory**

<b>9</b>	<b>Designing Active Directory Domain Services for Security. . . . .</b>	<b>241</b>
	The New User Interface. . . . .	241
	The New Active Directory Domain Services Installation Wizard . . . . .	243
	Read-Only Domain Controllers . . . . .	245
	Read-Only AD DS Database . . . . .	246
	RODC Filtered Attribute Set . . . . .	246
	Unidirectional Replication . . . . .	247
	Credential Caching . . . . .	247
	Read-Only DNS . . . . .	249
	Staged Installation for Read-Only Domain Controllers . . . . .	250
	Restartable Active Directory Domain Services . . . . .	251
	Active Directory Database Mounting Tool . . . . .	252
	AD DS Auditing . . . . .	254
	Auditing AD DS Access . . . . .	255
	Active Directory Lightweight Directory Services Overview . . . . .	258
	New Features in Windows Server 2008 for AD LDS . . . . .	261
	Active Directory Federation Services Overview . . . . .	261
	What Is AD FS? . . . . .	262
	What Is New in Windows Server 2008? . . . . .	263
	Summary . . . . .	264
	Additional Resources . . . . .	264

<b>10</b>	<b>Implementing Active Directory Certificate Services. . . . .</b>	<b>265</b>
	What Is New in Windows Server 2008 PKI. . . . .	266
	Threats to Certificate Services and Mitigation Options . . . . .	267
	Compromise of a CA's Key Pair. . . . .	267
	Preventing Revocation Checking. . . . .	268
	Attempts to Modify the CA Configuration. . . . .	271
	Attempts to Modify Certificate Templates . . . . .	272
	Addition of Nontrusted CAs to the Trusted Root CA Store . . . . .	273
	Enrollment Agents Issuing Unauthorized Certificates . . . . .	274
	Compromise of a CA by a Single Administrator . . . . .	275
	Unauthorized Recovery of a User's Private Key from the CA Database. . . . .	277
	Securing Certificate Services. . . . .	277
	Implementing Physical Security Measures . . . . .	278
	Best Practices. . . . .	279
	Summary . . . . .	280
	Additional Resources . . . . .	280

## **Part III Common Security Scenarios**

<b>11</b>	<b>Securing Server Roles . . . . .</b>	<b>285</b>
	Roles vs. Features . . . . .	286
	Default Roles and Features . . . . .	287
	Your Server Before the Roles. . . . .	294
	Default Service Footprint . . . . .	294
	Server Core . . . . .	294
	Roles Supported by Server Core . . . . .	296
	Features Supported by Server Core . . . . .	297
	What Is Not Included in Server Core. . . . .	297
	Tools to Manage Server Roles. . . . .	298
	Initial Configuration Tasks. . . . .	299
	Add Roles and Add Features Wizards . . . . .	299
	Server Manager . . . . .	300
	The Security Configuration Wizard . . . . .	302
	Multi-Role Servers . . . . .	311
	Summary . . . . .	312



<b>12</b>	<b>Patch Management</b>	<b>313</b>
	The Four Phases of Patch Management	313
	Phase 1: Assess	314
	Phase 2: Identify	315
	Phase 3: Evaluate and Plan	318
	Phase 4: Deploy	319
	The Anatomy of a Security Update	320
	Supported Command-Line Parameters	321
	Integrating MSU Files into a Windows Image File	321
	Tools for Your Patch Management Arsenal	322
	Microsoft Download Center	322
	Microsoft Update Catalog	322
	Windows Update and Microsoft Update	323
	Windows Automatic Updating	324
	Microsoft Baseline Security Analyzer	326
	Windows Server Update Services	330
	System Center Essentials 2007	338
	Summary	339
	Additional Resources	340
<b>13</b>	<b>Securing the Network</b>	<b>341</b>
	Introduction to Security Dependencies	344
	Acceptable Dependencies	345
	Unacceptable Dependencies	345
	Dependency Analysis of an Attack	347
	Types of Dependencies	348
	Usage Dependencies	349
	Access-Based Dependencies	349
	Administrative Dependencies	352
	Service Account Dependencies	352
	Operational Dependencies	352
	Mitigating Dependencies	353
	Step 1: Create a Classification Scheme	354
	Steps 2 and 3: Network Threat Modeling	357
	Step 4: Analyze, Rinse, and Repeat as Needed	360
	Step 5: Design the Isolation Strategy	361
	Step 6: Derive Operational Strategy	363
	Step 7: Implement Restrictions	363

	Summary .....	366
	Additional Resources .....	367
<b>14</b>	<b>Securing the Branch Office.....</b>	<b>369</b>
	An Introduction to Branch Office Issues .....	369
	Why Do Branch Offices Matter? .....	370
	What Is Different in a Branch Office? .....	370
	Building Branch Offices .....	371
	Windows Server 2008 in the Branch Office .....	373
	Nonsecurity Features .....	373
	Security Features for the Branch Office .....	376
	Other Security Steps .....	389
	Summary .....	390
	Additional Resources .....	390
<b>15</b>	<b>Small Business Considerations.....</b>	<b>391</b>
	Running Servers on a Shoestring .....	392
	Choosing the Right Platforms and Roles .....	393
	Servers Designed for Small Firms .....	395
	Windows Server 2008 Web Edition .....	395
	Windows Server Code Name “Cougar” .....	395
	Windows Essential Business Server .....	399
	Hosted Servers .....	400
	Virtualization .....	400
	Violating All the Principles with Multi-Role Servers .....	401
	Acceptable Roles .....	402
	Server Components .....	402
	Risk Considerations .....	403
	Edge Server Issues .....	405
	Supportability and Updating .....	406
	Server Recoverability .....	407
	Best Practices for Small Businesses .....	409
	Following Hardening Guidance .....	409
	Policies .....	413
	Vendor Best Practices .....	415
	Remote Access Issues .....	417
	Monitoring and Management Add-ons .....	418
	The Server’s Role in Desktop Control and Management .....	420
	Recommendations for Additional Server Settings and Configurations .....	423

Summary . . . . .	428
Additional Resources . . . . .	428
<b>16 Securing Server Applications . . . . .</b>	<b>431</b>
Introduction . . . . .	431
IIS 7: A Security Pedigree . . . . .	433
Configuring IIS 7 . . . . .	433
Feature Delegation . . . . .	434
TCP/IP-Based Security . . . . .	436
IP Address Security . . . . .	436
Port Security . . . . .	438
Host-Header Security . . . . .	439
Simple Path-Based Security . . . . .	439
Defining and Restricting the Physical Path . . . . .	440
Default Document or Directory Browsing? . . . . .	443
Authentication and Authorization . . . . .	444
Anonymous Authentication . . . . .	445
Basic Authentication . . . . .	446
Client Certificate Mapping . . . . .	447
Digest Authentication . . . . .	450
ASP.NET Impersonation . . . . .	451
Forms Authentication . . . . .	451
Windows Authentication . . . . .	452
Trusting the Server . . . . .	453
Further Security Considerations for IIS . . . . .	455
Summary . . . . .	460
Additional Resources . . . . .	461
<b>Index . . . . .</b>	<b>463</b>

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

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

[www.microsoft.com/learning/booksurvey/](http://www.microsoft.com/learning/booksurvey/)

# Understanding User Account Control (UAC)

— Darren Canavor

## In this chapter:

What Is User Account Control? .....	92
How Token Filtering Works .....	92
Components of UAC .....	94
UAC Group Policy Settings .....	108
What's New in UAC in Windows Server 2008 and Windows Vista SP1 .....	111
UAC Best Practices .....	112
Summary .....	113
Additional Resources .....	114

With a shift in the way people use computers, such as performing banking transactions, making online purchases, and sharing and storing personal information, a new set of security threats emerged. Windows users were largely running with administrative privileges all the time. If the user accidentally installed malicious software (malware) onto such a computer, that malware—which had administrator access—could do anything. In Windows Vista and Windows Server 2008, the new User Account Control (UAC) feature is designed to apply the principle of “least privilege”: Only give enough access to perform the task with as few disruptions as possible to the user experience. That includes all interactive users, with the exception of the built-in Administrator account. This may sound simple, but the challenge required a solution encompassing extensive changes to the core operating system, changes in industry perception of the standard user desktop and broad adoption of standard user best practices by the independent software vendor (ISV) community.



**Note** Although UAC is available in Windows Server 2008, it is primarily considered a client feature. To a systems administrator, the impact of UAC on Windows Server 2008 focuses on using Group Policy to manage Windows Vista client UAC policies.

# What Is User Account Control?

UAC can help prevent unauthorized changes to a computer by allowing the user to verify actions before they happen. When a user designated with elevated privilege logs on to Windows Vista and Windows Server 2008, two access tokens are issued: a full access token and a filtered standard user access token. The filtering process removes the administrative privileges, and disables the Administrative group Security Identifiers (SIDs), resulting in a filtered standard user access token. The standard user token is then used to start the Windows desktop (explorer.exe) and all subsequent child processes. Consequently, all applications run with the standard user token by default, and only when granted permission by an administrator will the application run with a full access token. Note that because applications inherit the privilege level of the parent process, if the parent process is running with a full access token, the new child process will inherit and run without prompting the administrator for permission. For example, if you launch a command prompt as an administrator, any process you launch from within the command prompt will run as an administrator.



**On the CD**   Elevating Explorer

By default Explorer.exe is designed not to be elevated. Consequently, if you right-click the binary and select Run As Administrator it will launch a new window, but in the same context as the original. On the companion CD, you will find a set of elevation tools, including a tool that puts an Elevate Explorer Here item on the right-click menu of any folder. Using that tool, you can launch an elevated Windows Explorer instance anywhere you wish.

# How Token Filtering Works

When a user logs on to a Windows Vista or Windows Server 2008 computer, the operating system examines the Relative IDs (RIDs) and privileges of the user. The user will receive two tokens (filtered and full) if her account possesses any of the RIDs listed in Table 4-1 or any of the privileges listed in Table 4-2.

**Table 4-1   UAC List of Restricted RIDs**

Restricted RIDs	Description
DOMAIN_GROUP_RID_ADMINS	Administrative domain user account
DOMAIN_GROUP_RID_CONTROLLERS	Domain Controllers group
DOMAIN_GROUP_RID_CERT_ADMINS	Certificate Publishers group
DOMAIN_GROUP_RID_SCHEMA_ADMINS	Schema administrators group
DOMAIN_GROUP_RID_ENTERPRISE_ADMINS	Enterprise Administrators group
DOMAIN_GROUP_RID_POLICY_ADMINS	Policy Administrators group
DOMAIN_ALIAS_RID_ADMINS	Administrative local user account
DOMAIN_ALIAS_RID_POWER_USERS	Power Users group
DOMAIN_ALIAS_RID_ACCOUNT_OPS	Account Operators group, Server only

**Table 4-1 UAC List of Restricted RIDs**

Restricted RIDs	Description
DOMAIN_ALIAS_RID_SYSTEM_OPS	System Operators group, Server only
DOMAIN_ALIAS_RID_PRINT_OPS	Print Operators group, Server only
DOMAIN_ALIAS_RID_BACKUP_OPS	Backup Operators group
DOMAIN_ALIAS_RID_RAS_SERVERS	RAS and IAS servers group
DOMAIN_ALIAS_RID_PREW2KCOMPACCESS	Pre-Windows 2000 Compatibility Access group
DOMAIN_ALIAS_RID_NETWORK_CONFIGURATION_OPS	Network Configuration Operators group
DOMAIN_ALIAS_RID_CRYPTO_OPERATORS	Cryptographic Operators group

**Table 4-2 UAC List of Restricted Windows Privileges**

Restricted Windows Privileges	Description
SeCreateTokenPrivilege	Required to create a primary token
SeTcbPrivilege	Identifies holder as part of the trusted computing base
SeTakeOwnershipPrivilege	Take object ownership without being granted discretionary access
SeBackupPrivilege	Required to perform system-wide backup tasks
SeRestorePrivilege	Required to perform system-wide restore tasks
SeDebugPrivilege	Can debug the memory of a process owned by another account
SeImpersonatePrivilege	Required to impersonate a client after authentication
SeRelabelPrivilege	Required to modify an object's mandatory integrity level

The filtered standard user token will have all Windows privileges removed except the list of standard Windows privileges shown in Table 4-3.

**Table 4-3 UAC List of Standard Windows Privileges**

Standard Windows Privileges	Description
SeChangeNotifyPrivilege	Required to receive file or folder change notifications
SeShutdownPrivilege	Required to shut down a system remotely
SeUndockPrivilege	Required to undock a laptop
SeReserveProcessorPrivilege	Required to modify user processor privilege
SeTimeZonePrivilege	Required to adjust the computer's time zone

The filtered access token has all the RIDs from Table 4-1, if present, marked as `USE_FOR_DENY_ONLY`. It also has the privileges listed in Table 4-2 removed. The unmodified full administrator access token is linked to the filtered access token and is used when requests are made to launch applications with a full administrator access token.

You can find more information on RIDs in Chapter 1, “Subjects, Users, and Other Actors.” You can find more information on Windows privileges in Chapter 3, “Objects: The Stuff You Want.”

## Components of UAC

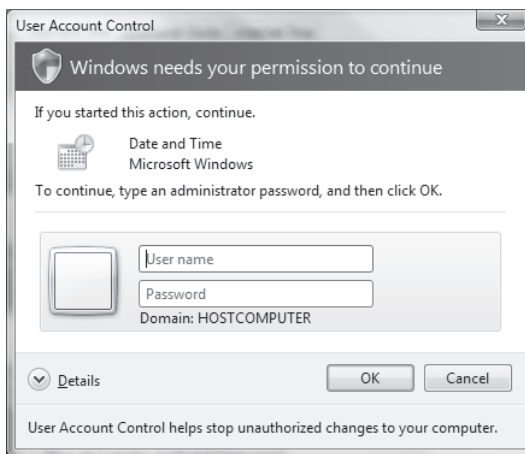
UAC is primarily perceived to be the elevation prompt. However, although that part is the most visible, it is not the most important part of UAC. UAC, in fact, consists of a number of components, all of which contribute in some way to enabling more people to run as nonadministrators, which is the ultimate goal of UAC. This section discusses the various components of UAC, starting with the various types of elevation dialogs.

### UAC Elevation User Experience

The most salient impact UAC has on user experience will be seen by users who are members of the local administrator group. Standard users also have the ability to perform administrative tasks without having to log off. The prompt for standard users is identical to the administrative prompt, except it requires password entry.

#### The Credential Prompt

On Windows Vista and Windows Server 2008, with the exception of the built-in administrators account, all users start applications without administrator-level privilege. When a given task requires administrator privilege the interactive standard user will be presented with an elevation credential prompt, shown in Figure 4-1, requiring the entry of a valid user name and password of a user that is a member of the Local Administrators group.

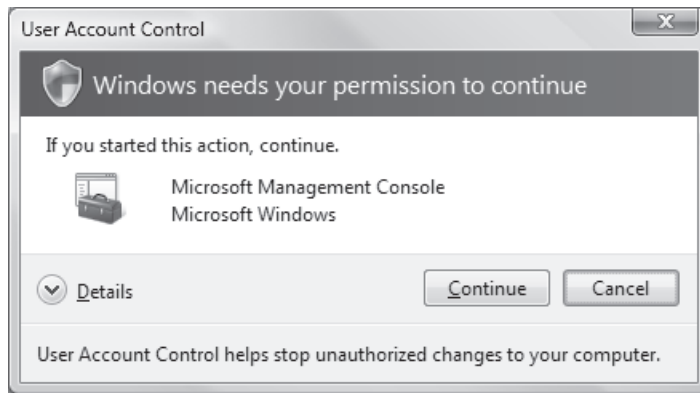


**Figure 4-1** A standard user is presented with a prompt for credentials when attempting to perform an administrative action.

#### The Consent Prompt

By default the consent prompt, shown in Figure 4-2, is presented when a user who is a member of the local administrators group attempts to perform a task that requires administrator

privilege. This consent prompt is presented only to local administrators running in Admin Approval Mode.



**Figure 4-2** An administrator is presented with a prompt for consent when attempting to perform an administrative action.

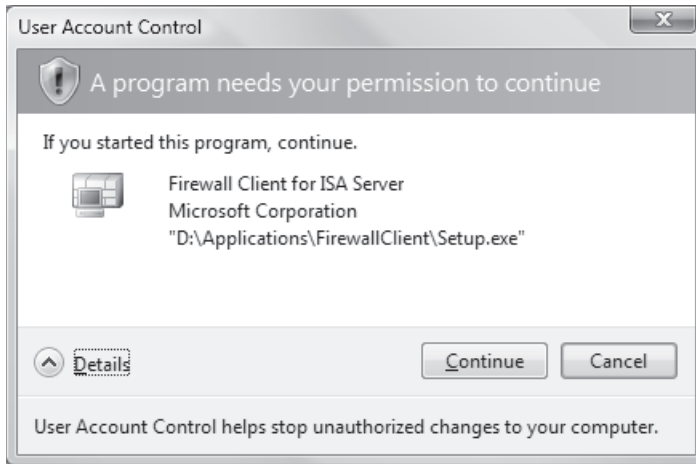
To help users make informed decisions, the UAC elevation prompts are color-coded and use different text to indicate an application's potential security risk. For example, the color (four-color shield on blue-green bar) and text of Figure 4-2 indicate a Windows Vista or Windows 2008 application requiring administrative access, such as the Microsoft Management Console.

When an application attempts to run with an administrator's full access token, Windows Vista and Windows Server 2008 analyze the executable to determine its publisher and uses this information to determine the correct user experience.

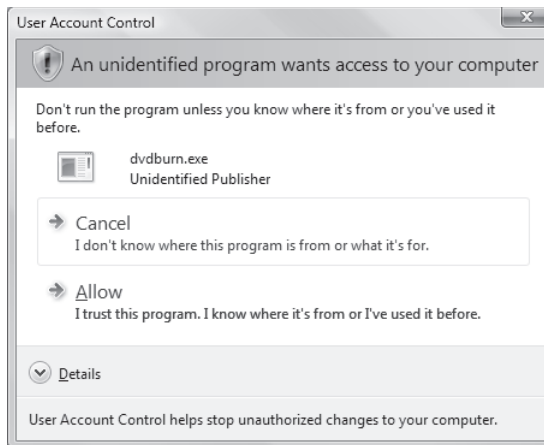
Various alternative prompts are shown in Figures 4-3 through 4-5 and are distinguished by different colors and text. For example, in Figure 4-3 the color (yellow shield on gray bar) and text indicate that the application requiring administrative access is Authenticode signed and trusted by the local computer, such as the Microsoft Firewall Client for ISA Server. In Figure 4-4 the color (yellow shield on yellow bar) and text indicate that the application requiring administrative access is unidentified and does not have a valid Authenticode signature from the publisher; therefore, take care before permitting the application to run. And in Figure 4-5 the color (red shield on red bar) and text indicate that the application requiring administrative access is from an explicitly blocked or untrusted publisher. An administrator can place the Publishers signing certificate in the local computer Untrusted certificate store to block a given publisher—this can also be set via Group Policy.

Note that UAC dialog boxes also change the displayed executable name and path details based on the trust level of the publisher's Authenticode signature. For example, in Figures 4-3 and 4-5, the user is trying to start the same application. The difference is that in Figure 4-3, the publisher is trusted, while in Figure 4-5 the publisher is explicitly blocked. When a publisher is trusted, not only does the dialog box color change, but the displayed text is also much friendlier.





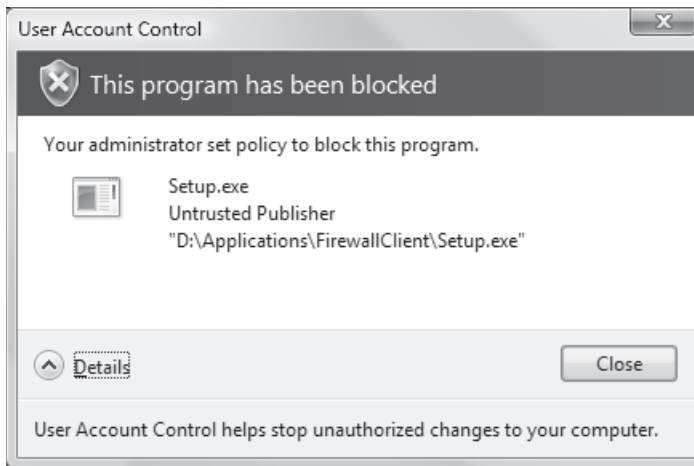
**Figure 4-3** UAC prompt indicating that the application requiring administrative access is Authenticode-signed and trusted by the local computer.



**Figure 4-4** UAC prompt indicating that the application requiring administrative access is unidentified and does not have a valid Authenticode signature from the publisher.

In Windows Vista and Windows Server 2008 the shield icon shown in Figure 4-6 denotes that when a user clicks a shielded control or program, UAC will prompt for authorization before continuing.

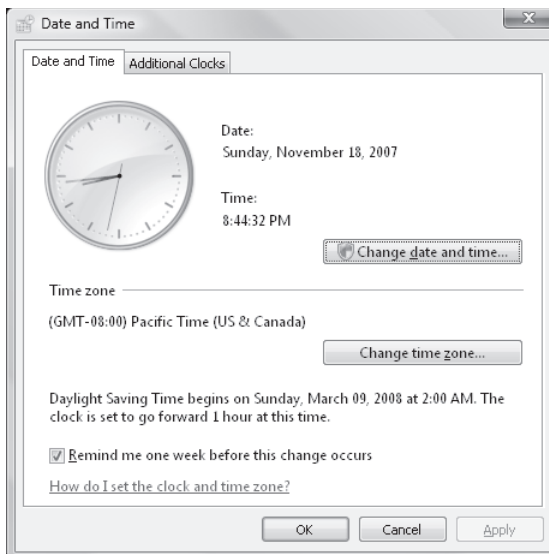
Some Control Panel components, such as the Date and Time, contain both administrator and standard user operations. For example, standard users can view the clock and change the time zone, but a full administrator access token is required to change the local system time, as shown in Figure 4-7. One reason for this is that a user who changes the system time can reorder events in the event log or impact the ability for a computer to authenticate to a Windows domain.



**Figure 4-5** UAC prompt indicating that the application requiring administrative access is from an explicitly blocked or untrusted publisher.



**Figure 4-6** The shield icon denotes an administrative action in Windows Vista and Windows 2008.



**Figure 4-7** The Date And Time Control Panel utility is used to configure local computer time and time zone.

## Application Information Service

The Application Information Service (AIS) is a new system service in Windows Vista and Windows Server 2008 that controls the launching of programs that require one or more elevated privileges, restricted rights, or privileged integrity levels to run. AIS is the component that actually launches these processes and attaches the right token to them. You could say that AIS is the heart of UAC. Note that AIS is disabled in Safe Mode; therefore, users who are members of the local administrator's group log on with their full administrative tokens. Windows took this approach because of the recovery and maintenance nature of Safe Mode scenarios.

## File and Registry Virtualization

Windows Vista and Windows Server 2008 include file and registry virtualization, which is a new application compatibility technology to address issues encountered by applications that historically required an administrator's access token to run. Virtualization helps mitigate these applications without burdening the ISV to make changes. A large number of legacy applications that previously failed to run without the administrators access token now work on Windows Vista and Windows Server 2008, thanks to virtualization.

When a legacy application running with a filtered standard user access token attempts to write to a protected directory, such as Program Files, the application is given a virtualized view of the resource it is attempting to change. The virtualized copy is maintained under the user's profile (or registry). Each user has a completely separate copy of the virtualized file. This means that two users playing the same game on the same computer may not see the same list of high scores, because each user could have his or her own virtualized vision of the game's %PROGRAMFILES%\Game\highscores.txt file. Therefore, IT administrators must understand file and registry virtualization and may potentially need to implement custom virtualization settings within the enterprise to overcome application compatibility issues. The following section examines file and registry virtualization.

### File Virtualization

File virtualization addresses the situation in which an application relies on creating or modifying files, such as a configuration file, in a protected location (%PROGRAMFILES%, %PROGRAMDATA%, or %SYSTEMROOT%) writeable only by administrators. Running such a program with a filtered standard user token may result in unexpected failures, or in some cases might be entirely blocked from running because of insufficient file or registry access.

When a program writes to a protected system location, the file virtualization filter driver (%SYSTEMROOT%\System32\Drivers\Luafv.sys) “traps” the operation and redirects it to a per-user location under the Virtual Store directory, located at %LOCALAPPDATA%\VirtualStore. When the program later reads the file, Luafv.sys traps the operation and again redirects it to the user's Virtual Store. If the file is not found in the Virtual Store, Luafv.sys will query the nonvirtualized location. Because file virtualization happens automatically, the

program believes it was successful in writing to %PROGRAMFILES%\appName. For security reasons file virtualization by default will not allow the redirection of known executable file types such as .exe, .dll, .sys, .bat, and .cmd. If, because of application compatibility constraints, the program needs to virtualize a .bat file, you can reconfigure the file virtualization filter to support this. The following examples demonstrate how to configure file virtualization.

**Configuring file virtualization to improve application compatibility** The FileList registry is not present by default and must be manually created to configure file virtualization.

Scenario: An enterprise relies on a legacy accounting application that writes a log file back to the application's restricted program folder. To enable virtualization on the accounting program's folder C:\appNameX, create a new DWORD named **Exclude** with a value of 0 under the following registry key:

```
[HKLM\SYSTEM\CurrentControlSet\Services\luafv\Parameters\FileList\Device\HarddiskVolume1 \appNameX]
```

Scenario: An enterprise forces all users to save their data to a specific location by locking down all user-writeable locations except the designated backup location. With virtualization enabled, a user can potentially store data in any virtualization-enabled location. To disable virtualization on a specific folder C:\Program Files\appNameY, create a new DWORD named **Exclude** with a value of 1 under the following registry key:

```
[HKLM\SYSTEM\CurrentControlSet\Services\luafv\Parameters\FileList\Device\HarddiskVolume1\Program Files\appNameY]
```

Scenario: An enterprise relies on a legacy accounting application that happens to write a .bat file back to the application's restricted program folder. To enable virtualization of .bat file extension types, create a new REG\_MULTI\_SZ named **ExcludedExtensionsRemove** with a value of bat under the following registry key:

```
[HKLM\SYSTEM\CurrentControlSet\Services\luafv\Parameters]
```



**Note** To expose virtual files and folders, browse to the virtualized file location using Windows Explorer and click Compatibility Files on the Explorer toolbar.

## Registry Virtualization

Registry virtualization is similar to file virtualization but applies to registry keys under HKLM\SOFTWARE. This feature permits applications that rely on the ability to store configuration information in HKLM\SOFTWARE to continue to operate when running without administrative privilege. The keys and data are redirected to HKEY\_CLASSES\_ROOT\

VirtualStore\SOFTWARE. Note that the VirtualStore location is created on demand by the first application utilizing virtualization. As with file virtualization, each user has a virtualized copy of values that an application has stored in HKLM. If, because of application compatibility constraints, a program needs to configure registry virtualization, this is supported. The following examples demonstrate how to configure registry virtualization.

## Configuring Registry Virtualization to Improve Application Compatibility

Scenario: An enterprise wants to prevent the virtualization of registry values under the key *DontVirtMe*. To do so, run the following command from an elevated command prompt:

```
Reg.exe flags HKLM\Software\DontVirtMe SET DON'T_VIRTUALIZE
```

Scenario: An enterprise wants to prevent the virtualization of all registry values and subkey values under the parent registry key *DontVirtMe*. To do so, run the following command from an elevated command prompt:

```
Reg.exe flags HKLM\Software\appName RECURSE_FLAG DONT_VIRTUALIZE
```

Although virtualization allows the overwhelming majority of pre-Windows Vista applications to run, it is a short-term fix rather than a long-term solution. In addition, some applications cannot be fixed, including applications that contain specific checks for user privileges. For example, many process-control applications check whether the user is an administrator, and exit if the user is not. You can get those applications to run on Windows Vista by attaching an application manifest that states the application needs to be run with administrative privileges and redeploy. Developers should modify all applications to comply with the Windows Vista and Windows Server 2008 Logo Program rather than relying on file and registry virtualization.

## Manifests and Requested Execution Levels

Applications running on Windows Vista and Windows Server 2008 can use application manifests to describe or declare requirements to the operating system at run time.

Administrative applications can declare their privilege requirements in the application manifest and the system will prompt the user for permission accordingly. Most pre-Windows Vista administrative applications, however, can run smoothly without modification even though they lack an entry in the application manifest. This is due to the vast array of Windows Vista and Windows Server 2008 application compatibility fixes, most of which depend on UAC being enabled. Application compatibility fixes enable applications to run that would normally fail if they ran without administrative access. For example, imagine a game that checks during start-up to see whether the user is a member of the local administrators group. Running with a filtered standard user access token, this check will fail—causing the application to fail. Using the application compatibility database, the operating system can discover that the application must run with a full token and prompt

the user accordingly or discover that the application runs fine without a full token and makes the application perceive it was started with a full token. These types of application compatibility fixes are called *shims*.

All Windows Vista and Windows Server 2008 logo-compliant applications must have a valid manifest with a defined requested execution level. The application uses the *requestedExecutionLevel* attribute to declare its access requirements. If the application requires administrative access, the application manifest specifies a requested execution level of *requireAdministrator*. This will ensure that the system identifies this program as an administrative application and provide the necessary elevation experience. Note that an application can also have mixed functionality—administrative and standard user—depending on the user. For example, the Microsoft Management Console (MMC) is marked *highestAvailable*. If a standard user runs the MMC, it will start with standard user privilege and will not prompt. If the user has a filtered access token, such as a local administrator or network operator, the operating system will prompt the user to launch MMC with the user's highest available privilege, allowing the administrator to have a different level of access than the network operator and the standard user.

## Installer Detection Technology

Installation programs are applications designed to deploy software, and most write to system directories and machine registry keys. These protected system locations typically require administrator-level privilege, which means that standard users do not have sufficient access to install most programs. Windows Vista and Windows Server 2008 heuristically detect installation programs, updaters, and uninstall programs that require administrator access to run. Installer detection is a key component of the UAC design. It facilitates the correct elevation experience and prevents installations from being executed without the user's knowledge.

Installer detection only applies to the following:

- 32-bit executables
- Applications without a *requestedExecutionLevel*
- Interactive processes running as a standard user with UAC enabled

The operating system will heuristically determine whether an application is an installer. Heuristics are based on the following attributes:

- Keywords included in the filename, such as *install*, *setup*, *update*, and other language equivalents
- Keywords in the following Versioning Resource fields of the executable: Vendor, Company Name, Product Name, File Description, Original Filename, Internal Name, and Export Name

- Keywords in the side-by-side manifest that are embedded in the executable
- Keywords in specific *StringTable* entries that are linked in the executable
- Key attributes in the RC data that are linked in the executable

For example, if you have an application called `setup.exe` or `install.exe`, it will be detected as an installer and will automatically get a prompt. You can find general information and an overview of the Microsoft Windows Installer at MSDN: <http://go.microsoft.com/fwlink/?LinkId=30197>.

## User Interface Privilege Isolation

User Interface Privilege Isolation (UIPI) is a new technology in Windows Vista and Windows Server 2008 to help isolate administrator-level processes from processes running with lower privileges on the same interactive desktop. UIPI prevents a lower-privilege application from using Windows messages to send input to a higher-privilege process. Sending input from one process to another allows a process to “inject” input into another process without the user providing consent.

UIPI defines a set of permitted Windows messaging interactions controlled by the highest of the different process levels. Higher privilege levels can send Windows messages to applications running at lower levels, but lower levels cannot send certain Windows messages to application windows running at higher levels. UIPI does not interfere or change the behavior of window messaging between applications at the same privilege level. UIPI comes into play for a user who is a member of the administrators group and chooses to run both administrator and standard user privileged applications on the same interactive desktop.

## Secure Desktop Elevation Prompts

Credential and consent prompts are displayed on the secure desktop by default in Windows Vista and Windows Server 2008. Every application must run on a desktop, and each interactive user receives a desktop upon logon where all her applications run. The Secure Desktop is used by the operating system for services and sensitive user interfaces such as the log-on interface.

By presenting the elevation prompt on the secure desktop, the operating system guarantees that the information being presented cannot be tampered with. When an executable requests elevation, the user is switched from the user’s interactive desktop to the secure desktop. The secure desktop renders a dimmed background of the user desktop and displays a highlighted elevation prompt. When the user clicks Continue or Cancel, the desktop automatically switches back to the user’s interactive desktop. While malware can paint over the interactive desktop and present an imitation of the secure desktop (spoofing), authorizing consent does not allow the malware elevation. If UAC is configured to prompt for credentials, malware imitating the credential prompt may gather the user’s credentials; however, the malware will be unable to use those credentials remotely to obtain administrator privilege. Somewhat

bizarrely, the malware will gain absolutely nothing from spoofing the admin approval mode dialog box. Malware cannot enter the user name and password into a valid UAC dialog box presented on the Secure Desktop, nor can it use `runas.exe` to invoke a process with elevated privilege or automate a legitimate UAC dialog box.

## Using Remote Assistance

In Windows Vista and Windows Server 2008, a domain user can run as a standard user and have a centralized IT group provide all administration tasks. Microsoft provides both Remote Desktop (RD) and Remote Assistance (RA) access to computers for different administration purposes. RD sessions are useful when an administrator does not require end-user interaction but does require full control of the remote computer. RA is useful for diagnosing and troubleshooting problems when the end user needs to demonstrate the problem to an IT expert. RA has been impacted by UAC; it is important that you understand how.

IT experts will experience two typical problems using RA. The first is that by default, the UAC prompts use the secure desktop and consequently are not available to the remote user. The second is if the UAC enterprise policy Behavior Of The Elevation Prompt For Standard Users is configured to Automatically Deny Elevation Requests, elevation is blocked entirely.

Windows Vista SP1 has a new UAC policy to address the challenge of the secure desktop prompting: User Account Control: UIAccess Applications To Prompt For Elevation Without Using The Secure Desktop. With this policy configured, AIS dynamically disables secure desktop prompting for UIAccess accessibility applications such as Remote Assistance and re-enables it once the program exits. For more details, see “What Is New in UAC in Windows 2008 and Windows Vista SP1” later in the chapter.

If the policy Behavior Of The Elevation Prompt For Standard Users is set to Automatically Deny Elevation Requests, the IT expert who connects using RA will be unable to launch an application with administrative privilege. To work around this issue, the IT expert can use `runas.exe` to launch a Command Prompt window using her own user name and password and then start a process that requires elevation. UAC will use the IT expert’s UAC prompt policy.

The following procedure could be used by an IT expert for running the Registry Editor with administrator privilege:

1. Open a command prompt and type **`runas /user:domain\ITExpert cmd.exe`**.
2. In the new Command Prompt window that opens up, type **`regedit.exe`**.
3. Respond to the UAC elevation prompt.

## UAC Remote Administrative Restrictions

When an administrator logs on to a Windows Vista or Windows Server 2008 computer remotely, using normal Windows networking, he logs on in Admin Approval mode, just as if he were logging on locally. To augment this behavior, UAC restricts remote administration to



prevent admin loopback attacks and help protect against local malicious software running remotely with administrative privilege. For example, admin loopback would occur when a user logs on with a filtered access token and then malware simply performs a **net use \\127.0.0.1\c\$** to obtain administrative access to the file system. When UAC remote restrictions are enabled, the loopback would also obtain a filtered access token and not full administrative access. This behavior works differently for different types of user accounts, as described in the following sections.

## Local User Accounts

Imagine that a user who is local to the server and a member of the local Administrators group on the server establishes a remote connection by **net use \* \\server\share**. In this scenario, the token used for that user on the server will not be a full administrative token as in previous versions of Windows. The user has no elevation potential on the remote computer and cannot perform administrative tasks. If the user wants to administer the workstation with a local account, she must interactively log on to the remote computer by Remote Assistance or Remote Desktop if available.

## Domain User Accounts

When a user with a domain user account logs on to a computer remotely, and he is a member of the local Administrators group, the domain user will run with a full administrator access token on the remote computer and UAC will not be in effect.

## Managing UAC Remote Restrictions

To disable UAC remote restrictions for local accounts and obtain Windows XP and Windows 2003 parity, create a DWORD named **LocalAccountTokenFilterPolicy** with a value of 1 under the following registry key:

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
```

## Mapping Network Drives When Running in Admin Approval Mode

When an administrator in Admin Approval mode maps a network share, that share is only associated to the current log-on session for the current process access token. This means that if a user running a command prompt (cmd.exe) with a filtered access token explicitly maps a network share, that network share would not be exposed to any elevated cmd.exe instances running with a full administrator access token. Note that only in the case of UNC paths will the sessions be automatically linked by the system.

You can configure a registry value to share network connections between processes started with the filtered access token and full access token for a member of the Administrators group only. When you enable this registry setting, if a network resource is mapped to an access token, the LSA checks whether another access token is associated with the current user

session. If the LSA determines that there is a linked access token, it adds the network share to the linked location.

To enable a linked network drive, create a DWORD named **EnableLinkedConnections** with a value of 1 under the following registry key:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System

### Direct from the Field: Which Accounts Are Accepted for Elevation

Recently I was asked to troubleshoot some elevation problems for a friend of mine. She was unable to elevate to change some networking parameters on her laptop. The laptop was domain-joined, but the DC was unavailable at the time. After a few minutes of troubleshooting, I wrote up the following scenario, which I think helps highlight how UAC is not always intuitive, as well as how it interoperates with other features of Windows:

- The computer is called Denise-PC.
- The computer is joined to example.com.
- The DC for example.com is offline—in other words, Denise-PC is roaming.
- She has only previously logged on to Denise-PC using EXAMPLE\Denise.
- EXAMPLE\Denise is a member of BUILTIN\Users.
- BUILTIN\Administrators on Denise-PC contains BUILTIN\Administrator and DENISE-PC\Denise.
- When she attempts an administrative action she gets an elevation prompt asking for an admin account.

We have several options for how to elevate:

1. Attempt to elevate to BUILTIN\Administrator.
2. Attempt to elevate to EXAMPLE\Denise.
3. Attempt to elevate to EXAMPLE\Administrator.
4. Attempt to elevate to EXAMPLE\Foo, where Foo is a member of EXAMPLE\Domain Admins.
5. Attempt to elevate to DENISE-PC\Denise.

Option 1 will fail because BUILTIN\Administrator is disabled by default in Windows Vista as long as there is another local admin account. Because DENISE-PC\Denise is a local admin, and it is enabled, BUILTIN\Administrator is not available for use.

Option 2 will fail as well. EXAMPLE\Denise is only a member of users. It is not an admin and therefore you cannot elevate to it.

Option 3 will fail because although EXAMPLE\Administrator is a member (indirectly) of BUILTIN\Administrators, it has never logged on to Denise-PC. Because the computer is offline, authentication of domain accounts has to happen against the password verifier. (See Chapter 2, “Authenticators and Authentication Protocols,” for information on cached credentials.) Cached credentials exist only for accounts that have previously logged on interactively; therefore, we have nothing to verify EXAMPLE\Administrator against. It should also be pointed out that elevating to a domain administrator on a member workstation would be an extraordinarily bad idea. For more information on why, see Chapter 13, “Securing the Network.”

Option 4 fails for the same reason as Option 3.

Option 5 will succeed. DENISE-PC\Denise is a local account. Therefore, no cached credentials are necessary. It is a member of BUILTIN\Administrators, so it is legal to elevate to this account. And it is not disabled, so it can be actively used to log on with.

I have found this write-up very helpful in explaining to people which accounts can be used for UAC elevation, as well as the relationship between domain accounts, password verifiers, and UAC.

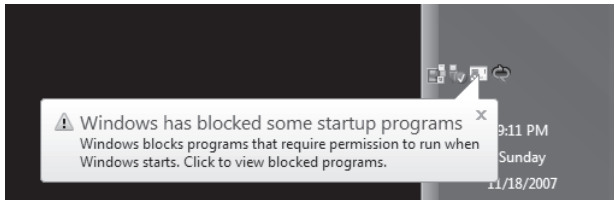
*Jesper M. Johansson*  
*Windows Security MVP*

## Application Elevations Blocked at Logon

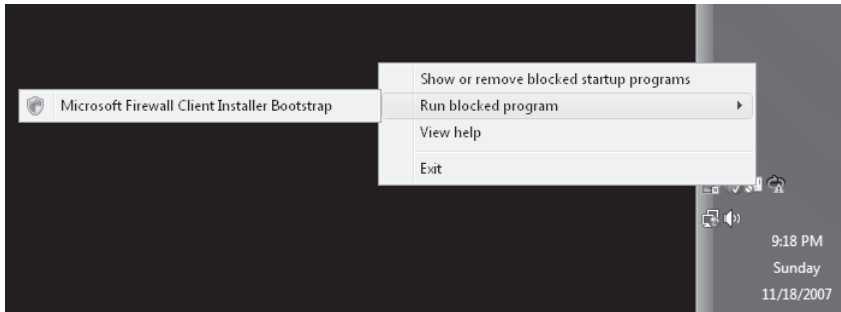
Windows Vista and Windows Server 2008 block administrative applications that try to start in the user’s log-on path. Many ISVs place programs in the user’s log-on launch path to ensure that they run each time the user logs on. While this solution may be convenient, it often results in application compatibility problems when the user logging on is not an administrator but the application requires him to be. This behavior is also convenient for malware, which can simply place itself in the user’s log-on launch location. From that point forward, every time the user logs on the malware runs silently with administrator-level access and without the user’s consent. To block this behavior, Windows Vista and Windows Server 2008 create a workflow to help the user manage the blocked list of programs. An elevation balloon notifies the user, as shown in Figure 4-8, and the tray icon allows the user to run the blocked program or enter the management UI, as shown in Figure 4-9.

With UAC, applications that require administrator-level privileges to run are blocked when launched from the following locations:

- Per-User Startup Folder %USERPROFILE%\Start Menu\Programs\Startup.
- Per-Machine Startup Folder %ALLUSERSPROFILE%\Start Menu\Programs\Startup.



**Figure 4-8** Blocked application balloon: Windows has blocked some start-up programs.



**Figure 4-9** Blocked application tray icon: Run Blocked Program/Show Or Remove Blocked Start-up Programs.

- Per-User RUN Key HKEY\_USERS\\*\Software\Microsoft\Windows\CurrentVersion\Run
- Per-Machine RUN Key HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.



**Note** In the preceding section, the asterisk ("\*") denotes all user security identifiers (SIDs) including the .Default SID.

It is important to note that enterprise Group Policy supports a user log-on script that will use the currently logged-on user's highest available access token; if the user is a member of the local administrators group, the script will elevate without prompting the Administrator in Admin Approval mode.

## Configuring Pre-Windows Vista Applications for Compatibility with UAC

The final and most important step in configuring UAC is ensuring that your software is either designed to be UAC compliant by following logo requirements or has been configured to run with Windows Vista or Windows Server 2008.

For new applications that are Windows Vista and Windows Server 2008 logo-compliant, the application must either run with a standard user privilege or, in the case of an administrative application, be marked with an application manifest entry. For more information, visit the Microsoft Windows logo home page at <http://www.microsoft.com/whdc/winlogo/hwrequirements.mspx>.

During the deployment of Windows Vista and Windows Server 2008, IT departments may discover some existing line-of-business (LOB) applications that will not function properly. In most cases, the problem is due to application incompatibility with the enhancements incorporated in the new operating systems. Microsoft provides an Application Compatibility Toolkit that assists in identifying the compatibility problems and aids in the creation of application compatibility fixes or shims. Some programs may need to perform administrative operations. For this to work correctly on Windows Vista and Windows Server 2008 under UAC, the program needs declare this to the operating system so that users will be prompted for approval before the application can run with a full administrator access token. The Application Compatibility Toolkit 5.0 with the Standard User Analyzer provides the means to test, build, and install the application compatibility database entries, which facilitate the requested execution level marking mechanism.

For information about application compatibility and the Application Compatibility Toolkit 5.0 featuring the Standard User Analyzer, visit TechNet at <http://go.microsoft.com/fwlink/?LinkId=23302>.

## UAC Group Policy Settings

The following section explores each of the eleven UAC group policies supported on Windows Vista and Windows Server 2008. These settings can be applied locally using the Local Security Policy editor or across an enterprise by using Group Policy.

## UAC Policy Settings Found Under Security Options

You can find the following nine UAC settings in the Group Policy Editor or the Local Security Policy editor under: Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options.

### User Account Control: Admin Approval Mode for the Built-in Administrator Account

This security setting controls the behavior of the Built-in Administrator (BA) account. If you use the BA account for daily administrative tasks, you may consider disabling this setting. However, if you do so, you also lose Internet Explorer Protected Mode. All applications will run as a full administrator. By default this policy is enabled.

## **User Account Control: Behavior of the Elevation Prompt for Administrators in Admin Approval Mode**

If an operation requires administrator privilege to start, this policy will control the UAC prompt experience for administrators in Admin Approval Mode. Although the default consent configuration is convenient, enforcing credentials may be desirable. For example, if a parent and child share the same user account, the child will be unable to perform elevated tasks without knowledge of the password. Also, in some cases administrators may want to disable elevation prompting without disabling UAC and therefore can set the elevation prompt to Silent. This retains Internet Explorer Protected Mode, but removes the elevation prompts. By default this policy is Prompt For Consent.

## **User Account Control: Behavior of the Elevation Prompt for Standard Users**

UAC provides an in-context elevation prompt experience, and if the user can provide a valid administrator user name and password, the elevated operation will succeed. For enterprises that do not want their users to have the opportunity to elevate, you can set this policy to automatically deny all elevation requests. By default this setting is Prompt For Credentials.

## **User Account Control: Detect Application Installations and Prompt for Elevation**

This setting enables or disables application installer detection. It is best to leave this setting enabled, which is the default.

## **User Account Control: Only Elevate Executables That Are Signed and Validated**

This setting will enforce Authenticode signature validation on any interactive applications requesting elevation. If an enterprise runs only Authenticode-signed programs, this setting can increase security by controlling which application publishers are allowed to run with elevated privileges. However, most users would experience significant application compatibility problems if they tried to use this setting, which is why it is disabled by default.

## **User Account Control: Only Elevate UIAccess Applications That Are Installed in Secure Locations**

UIAccess applications are most often accessibility programs that need to interact directly with the Windows UAC elevation dialogs. Windows Vista and Windows Server 2008 UAC elevation dialogs are protected with a high integrity level. For UIAccess applications to interact they must declare this requirement in the application manifest. When the program

starts, it receives a special integrity level permitting interaction. Because UIAccess applications are powerful, this setting enforces that such programs be started only from a secure directory file path. UIAccess applications must also have a valid and trusted Authenticode signature. By default this setting is enabled.

### **User Account Control: Run All Users, Including Administrators, as Standard Users**

This is the UAC on/off switch. Don't disable UAC! If UAC is disabled, all related features also become disabled. File and registry virtualization no longer function and all virtualized data appear lost to the user. Users who were running in Admin Approval Mode now log on with full administrative rights, and all applications run with administrator privilege, silently! Application compatibility shims designed to increase compatibility with pre-Windows Vista applications are also disabled. Internet Explorer's Protected Mode is disabled, forcing Internet Explorer to run with administrative privilege. Have we convinced you to leave UAC on? By default this setting is enabled.

### **User Account Control: Switch to the Secure Desktop When Prompting for Elevation**

This setting determines whether elevation requests are presented on the interactive user's desktop or on the secure desktop. The secure desktop prevents output spoofing, which means that whatever is presented on the secure desktop cannot be tampered with. UAC dialog boxes on the interactive user's desktop can be spoofed and therefore are less secure than those presented on the secure desktop. By default this setting is enabled.

### **User Account Control: Virtualize File and Registry Write Failures to Per-User Locations**

This setting enables or disables the redirection of write failures for the file system and registry. Disable this feature if you use only Windows Vista or Windows Server 2008 logo-compliant software. If you require custom virtualization settings, see "Configuring Registry Virtualization to Improve Application Compatibility" earlier in the chapter. By default this setting is enabled.

## **Related UAC policies**

Windows Vista and Windows 2008 also have two complementary policy settings: the Require Trusted Path For Credential Entry and Enumerate Local Administrator Accounts On Elevation settings. You can find both settings in the Group Policy Editor under: Local Computer Policy\Computer Configuration\Administrative Templates\Windows Components\Credential User Interface.

## Require Trusted Path for Credential Entry

This setting controls whether the user must enter Windows credentials using a trusted path. The trusted path is a secure key sequence—sometimes referred to as a Secure Attention Sequence (SAS)—which prevents malware from stealing your Windows credentials. When a standard user tries to perform a task requiring administrator privilege, the system forces the user to enter Ctrl+Alt+Delete before being redirected to the secure desktop to enter a valid administrator user name and password to complete the operation. This trusted path credential workflow prevents input spoofing and output spoofing, making this the most secure Windows credential input configuration. By default this setting is disabled.

## Enumerate Administrator Accounts on Elevation

This setting enables the automatic enumeration of local administrator accounts in the UAC credential UI, as shown in Figure 4-10. Note that in some domain-joined environments that encounter networking connectivity issues, this setting can cause unexpected delays when enumerating the local administrator accounts. By default this setting is disabled.

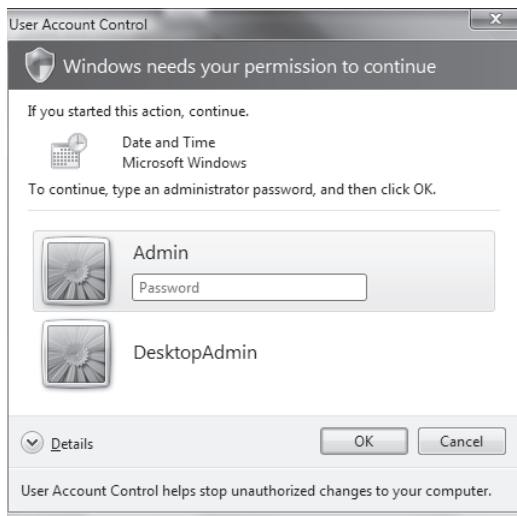


Figure 4-10 Automatic local administrator account enumeration in the UAC credential dialog box.

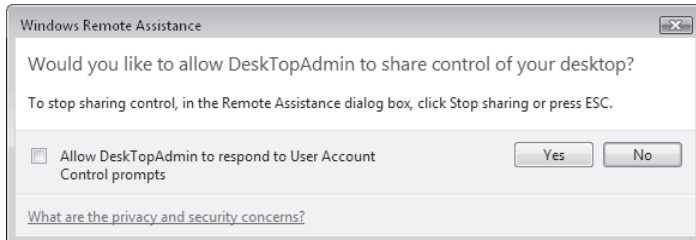
## What's New in UAC in Windows Server 2008 and Windows Vista SP1

UAC underwent only a few small changes in Windows Server 2008 and Windows Vista SP1 when compared to the original release in Windows Vista. The following sections summarize the UAC changes.



## New Group Policy Setting: UIAccess Applications to Prompt for Elevation without Using the Secure Desktop

This setting enables UIAccess applications such as Remote Assistance to request the disabling of secure desktop prompting. When the UIAccess application is complete, the secure desktop prompting is automatically enabled, thus removing the necessity for the end user to allow the desktop admin elevation access. (See Figure 4-11.) As discussed earlier in this chapter, this is a convenient setting for those enterprises that rely on Remote Assistance to provide end user desktop help desk support. By default this setting is disabled.



**Figure 4-11** Windows Remote Assistance: Allow Helpdesk To Respond To User Account Control Prompts.

## UAC Prompt Reduction When Performing File Operations in Windows Explorer

When a user creates a new folder in a protected location, the user will be prompted only once to create and name the folder. This was a two-prompt scenario in Windows Vista RTM.

## More Than 40 Additional UAC-Related Application Compatibility Shims

The UAC team in conjunction with the Application compatibility team produced over 40 new application shims to help increase Windows Vista and Windows 2008 compatibility.

## UAC Best Practices

Managing UAC is not as hard as it seems. How you deploy in an organization depends largely on your organization's security needs and tolerance to implement the required policies to meet those needs. The following solutions are presented in reverse order of preference (good, better, best) with respect to security value.

### Good Practice

Run users in Admin Approval Mode. If an administrative user requires elevated privileges, the enterprise UAC policy should enforce that the user enters a valid administrator user name and

password instead of simply clicking the Consent dialog box. This configuration prevents unauthorized elevations on the off-chance that a user leaves his workstation unattended. To improve security you could also require the Ctrl+Alt+Delete key sequence for any elevation to complete. This makes entering administrative credentials far more secure.

## Better Practice

Enforce that all users who require administrator privilege have two accounts: one standard user account for day-to-day activities such as reading e-mail and one for the occasional administrative operation. The standard user can log on and when needed can elevate using a UAC credential prompt. This is not the best solution because now the user is running both standard user and administrator-privileged applications in the same interactive session. To increase security an enterprise can enforce that the user must use Fast User Switching (FUS) anytime she needs to perform an elevated operation. Although FUS is more secure, it does have user experience drawbacks. To improve security you could also require the Ctrl+Alt+Delete key sequence as with the previous option.

## Best Practice

Run all users as standard users. The IT department must then assume that standard users will generally not be able to install applications and therefore must deploy software on their behalf. Windows provides an installation service to do this called the Microsoft Software Installer (MSI) Service. In addition, the Group Policy Software Installation (GPSI) extension allows applications to be distributed to a user's computer without any user interaction required. See the Group Policy Software Installation Extension documentation at <http://go.microsoft.com/fwlink/?LinkId=71356> for more information.

## Summary

UAC is probably the most talked-about feature in Windows Vista. It is even the subject of advertisements from rival software vendors. It is hard to say whether to be flattered or annoyed that Microsoft's competitors are now advertising their products as more desirable because Windows is too secure. Regardless, UAC is a critical step for Windows. The status quo, where users run as administrators to get normal tasks done, is unacceptable and has led to a malware pandemic. Only by helping users run as nonadministrators can we ever hope to stem the flood of malware and reduce desktop total cost of ownership (TCO).

The future is one where users only use administrative privileges where necessary. UAC is a step in that direction, but it will only work if people use it, and if they demand that their ISVs get software that works as a standard user. You can do your part in protecting the IT ecosystem by using UAC, and by buying software that works with it and rejecting software that does not.

## Additional Resources

- Microsoft Corporation (2006). “The Windows Vista and Windows Server 2008 Developer Story: Windows Vista Development Requirements for User Account Control (UAC),” at <http://msdn2.microsoft.com/en-us/library/aa905330.aspx>.
- Mark Russinovich (2007). “Inside Windows Vista User Account Control,” at <http://www.microsoft.com/technet/technetmag/issues/2007/06/UAC/>.
- Raymond Chen (2006). “An Administrator Is Not the Administrator,” at <http://www.microsoft.com/technet/technetmag/issues/2006/03/WindowsConfidential/?related=/technet/technetmag/issues/2006/03/WindowsConfidential>.
- Wole Moses (2007). “Services Hardening in Windows Vista,” at <http://www.microsoft.com/technet/technetmag/issues/2007/01/SecurityWatch/?related=/technet/technetmag/issues/2007/01/SecurityWatch>.