

Active Directory Field Guide



Laura E. Hunter

Apress®

Active Directory Field Guide

Copyright © 2005 by Laura E. Hunter

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN (pbk): 1-59059-492-4

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Jim Sumser

Technical Reviewer: Alexander N. Nepomnjashiy

Editorial Board: Steve Anglin, Dan Appleman, Ewan Buckingham, Gary Cornell, Tony Davis, Jason Gilmore, Jonathan Hassell, Chris Mills, Dominic Shakeshaft, Jim Sumser

Assistant Publisher: Grace Wong

Project Manager: Beckie Stones

Copy Manager: Nicole LeClerc

Copy Editor: Ami Knox

Production Manager: Kari Brooks-Copony

Production Editor: Ellie Fountain

Compositor: Diana Van Winkle

Proofreader: Linda Marousek

Indexer: Kevin Broccoli

Artist: Diana Van Winkle

Cover Designer: Kurt Krames

Manufacturing Manager: Tom Debolski

Distributed to the book trade in the United States by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013, and outside the United States by Springer-Verlag GmbH & Co. KG, Tiergartenstr. 17, 69112 Heidelberg, Germany.

In the United States: phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders@springer-ny.com, or visit <http://www.springer-ny.com>. Outside the United States: fax +49 6221 345229, e-mail orders@springer.de, or visit <http://www.springer.de>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.



Deploying Group Policy

Or How to Run the Whole Thing from Your Desk and Still Leave for Home by 5

I sometimes feel like a used car salesman when I start espousing the virtues of Group Policy, but I really can't help it; it's such a useful and complex tool that there seems to be very little that you can't do with it. (Okay, making au gratin potatoes might be a bit beyond its reach, but give the Redmond developers some time and I'm sure they'll figure it out.) Although it may sound overzealous of me to say, Group Policy might be one of the most useful tools that you can use as an Active Directory administrator. Group Policy integrates directly into Active Directory and allows you to manage and configure your servers and workstations from one single point. You can use Group Policy Objects (GPOs) to control almost every aspect of your computing environment, from creating a consistent desktop configuration, to securing your systems, to deploying and managing software across anything from a home office to a large enterprise. In this chapter I'll show you some of the more useful features of this technology, starting with a new tool that makes enterprise Group Policy management a snap. I'll then present some common scenarios that you might encounter as a consultant: using GPOs to create a consistent configuration for your desktop computers, and deploying software to the client desktop. I'll finish up with some advanced tips and techniques to use with Group Policy, including creating customized administrative templates and ways to exert granular control over how Group Policy Objects are deployed across your network.

In this chapter, we'll cover the following topics and tasks:

- Using the Group Policy Management Console
- Backing up, restoring, and migrating Group Policy Objects
- Creating Group Policy Modeling and Planning reports
- Using Group Policy to control client desktop configurations

- Implementing Software Restriction Policies
- Customizing security settings for your clients and workstations
- Using Group Policy to deploy software
- Controlling Group Policy deployment
- Advanced Group Policy tips and tricks

Group Policy Management Console

One of the reasons Group Policy doesn't get leveraged as much as it should is, I think, that the tools for managing it were a bit kludgy under Windows 2000, and even in the initial release of Windows Server 2003. Not anymore, though, since the Group Policy Management Console (GPMC) is *not* the same old clunky thing that you've been living with. But this tool has managed to fly under some people's radar because it wasn't released as part of the 2003 operating system. GPMC is an out-of-band product that you can download from the Microsoft website to manage Group Policies on 2003 networks. (You can even use GPMC on a 2000 AD domain, but some of the more advanced features won't be available to you.) GPMC can be installed on any Windows Server 2003 machine or XP workstation with Service Pack 1 or higher. Windows XP computers will also need the .NET Framework installed. (The GPMC will not run on 64-bit versions of Windows.) GPMC offers you the following key features to simplify Group Policy management:

- A simplified user interface that makes Group Policy much easier to use, manage, configure, and secure
- The ability to back up and restore individual Group Policy Objects, as well as all GPOs in a domain
- HTML reporting of the settings in an individual GPO
- Import/export and copy/paste functions for GPOs and Windows Management Instrumentation (WMI) filters

Note If you're on XP Service Pack 1, you'll need to have the Q326469 hotfix installed. But don't worry; the installer will let you know if it's not, and even install it for you automatically.

The installer itself is pretty straightforward: as long as you're on an XP or 2003 machine, you just double-click the .MSI file and click **Next** until you're done. The console will launch, attach to the domain that your computer is a member of, and provide a graphical view of your forest as you can see in Figure 4-1.

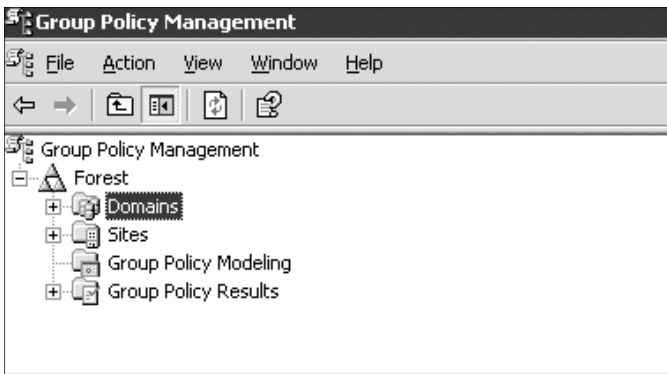


Figure 4-1. *The Group Policy Management Console*

Once you expand the Domains node, you'll see an entry for each Organizational Unit (OU) in your domain, and each GPO that's linked to a particular OU. Additionally, you'll see a list of every GPO that you've created, whether it's currently linked to a container or not. When you click an individual GPO, you can see a graphical summary of the settings it includes, like the one shown in Figure 4-2. You can view individual GPOs by drilling down in the following order:

1. Domains
2. *Your Domain Name*
3. Group Policy Objects

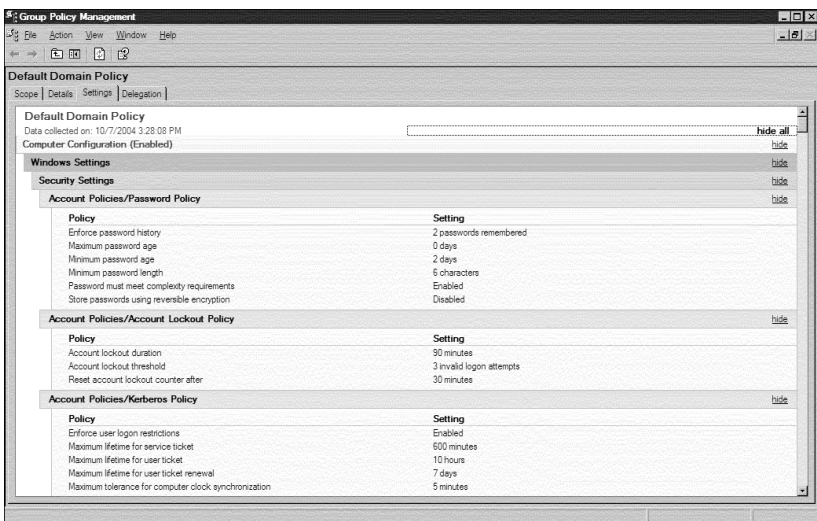


Figure 4-2. *GPMC settings report*

As you can see, this gives you an easy-to-read overview of your Group Policy settings. In addition, the **Scope** tab spells out precisely which domains, sites, and OUs are affected by this GPO and what Access Control Lists (ACLs) have been applied on it. Having this information right at your fingertips can save you a lot of time in troubleshooting, especially if you're taking over a network from someone else who may not have documented his GPO configurations very well.

Note While you're looking at the pretty reports that GPMC generates for you, why don't you take a second to right-click the GPO and click **Save Report**. This will save the information to an HTML file that you can use as a part of your network documentation, so that the person who takes over from *you* doesn't have the same complaint.

Another great feature of GPMC, and one that's firmly embedded in the “Why didn't they let us do this before?” column, is the ability to back up and restore your Group Policy Objects, and even copy GPOs between different domains and forests. Prior to the release of the GPMC, your only option for backing up a GPO before making a change was to manually create a copy of the object, which quickly became time-consuming and a waste of resources if you wanted to maintain multiple copies that you could roll back to. Now it's easy: just right-click the GPO that you want to back up and click **Back Up**. (Or click the top-level Group Policy node and click **Back Up All** to take care of all of them at once.) This will create a series of file folders in the location you specify, all with rather ugly-looking GUID names like this:

- C:\{03230347-3CC1-46BA-996C-2B4937757EEC}
- C:\{21B22D32-DA2F-40FF-AD12-4DB9F62271F5}
- C:\{41153A47-CB86-4090-8786-88EB9D110560}
- C:\{4C5857F4-4D78-4A26-902D-1038A3AE55AF}
- C:\{549ED06F-D275-473E-B944-5952DC354DC1}

Once you've created the backups, you can store them to tape as a part of your usual backup schedule. But it gets even better than that: the GPMC installation includes a Scripts folder (installed to C:\Program Files\GPMC\Scripts by default) with a number of predefined scripts that you can use to automate your administration tasks. BackupAllGPOs.wsf allows you to back up all of your GPOs within a given domain from the command line as a one-time or scheduled task, using the following syntax:

```
BackupAllGPOs.wsf BackupLocation [/Comment:value] [/Domain:value]
```

So a scheduled task to back up all of your GPOs to a file folder on your C:\ drive would look like this:

```
BackupAllGPOs.wsf c:\GPO-Backups /comment:"Back Up All Domain GPOs"
```

More than a dozen other predefined scripts are provided with the GPMC, including scripts that allow you to

- Back up all GPOs in your domain.
- Back up a single GPO.
- Find any disabled GPOs.
- Find GPOs with duplicate names.
- Get summary reports for all GPOs.

There's also a useful help file (found in the C:\Program Files\GPMC directory by default), called `gpmc.chm`, to get you started with these if you're unfamiliar with scripting. I'll admit that even *I* used to be immensely script-o-phobic, but getting past the script fear will make you a much better network manager. So say it with me, if you haven't already: "Scripting is my friend." We'll be talking about various other scripting solutions throughout this guide; it's a deceptively simple technology that will allow you to do pretty complex things with a relatively low learning curve.

Migrating Group Policy Settings

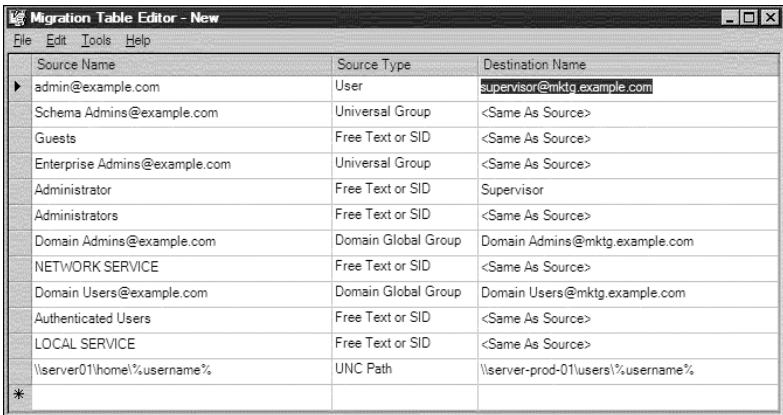
If you've ever wished that you could quickly move the settings from one Group Policy Object to another, the GPMC will also make that task immensely simple. As long as you have sufficient rights to both the source and destination domains and forests, copying a GPO is as simple as right-clicking a GPO and selecting **Copy** or **Import** within the console window. This is especially useful if you maintain a separate Active Directory forest for testing purposes; once you've perfected policy settings in the test area, you can simply copy the finished GPO into your production domain.

Note In a *copy* operation, a new GPO is created in the destination domain or forest. During an *import*, GPMC requires that the destination object already exist; the imported settings will overwrite any existing information in the destination GPO.

You'll also have access to a Migration Table during the copy process. This will let you map any domain-specific settings like usernames, SIDs, and UNC paths from the source domain into the target so that the copied

information will match up correctly. For example, you may have a test server called \\TEST-01 that contains the user directories for your test environment. But when you copy your GPO into production, you want \\TEST-01\HOME\%username% to change to \\APPI1\HOME\%username%; the Migration Table will allow you to do this without needing to make the change manually anywhere that it occurs within the Group Policy Object. To use the Migration Table, follow these steps:

1. Run **mtedit.exe** from the GPMC installation folder. This will create a blank Migration Table.
2. To automatically populate the Migration Table, click **Tools ► Populate from GPO** or **Tools ► Populate from Backup**.
3. By default, each source value will assume that its destination name is the same as the value listed for source name. It's up to you to manually edit the Destination Name column to include the appropriate values, as you can see in Figure 4-3.



Source Name	Source Type	Destination Name
admin@example.com	User	supervisor@mktg.example.com
Schema Admins@example.com	Universal Group	<Same As Source>
Guests	Free Text or SID	<Same As Source>
Enterprise Admins@example.com	Universal Group	<Same As Source>
Administrator	Free Text or SID	Supervisor
Administrators	Free Text or SID	<Same As Source>
Domain Admins@example.com	Domain Global Group	Domain Admins@mktg.example.com
NETWORK SERVICE	Free Text or SID	<Same As Source>
Domain Users@example.com	Domain Global Group	Domain Users@mktg.example.com
Authenticated Users	Free Text or SID	<Same As Source>
LOCAL SERVICE	Free Text or SID	<Same As Source>
\\server01\home\%username%	UNC Path	\\server-prod-01\users\%username%
*		

Figure 4-3. *Creating a Migration Table*

Note You can either type in the mapped entries manually, or right-click the Destination field and select **Browse**.

4. When you've finished, click **Tools ► Validate Table**, which verifies that any security principals and UNC paths in the Destination column actually exist.

Modeling Group Policy

The Group Policy Management Console also assists you in planning and troubleshooting your Group Policy strategy by allowing you to play out “What if?” scenarios using a modeling tool. Like the rest of the GPMC, Group Policy Modeling is fairly intuitive and wizard-driven; in fact, if you’ve used the Resultant Set of Policy (RSOP) Wizard or **gpresult.exe** from the Windows 2000 Resource Kit, you’re probably already familiar with the available options. To create a Group Policy Modeling report, follow these steps:

1. Open the Group Policy Management Console. Right-click Group Policy Modeling and select **Group Policy Modeling Wizard**. Click **Next** to bypass the initial **Welcome** screen.
2. Select the domain and domain controller that you want to use to perform the test. The DC needs to be a 2003 DC: you can either select a specific DC or allow the wizard to pick any available 2003 box. Click **Next** when you’re done.
3. Select 1) the user or user container, and 2) the computer or computer container that you want to analyze. You can mix-and-match these as well: you can pick a single user object and an OU that contains your computer accounts, a single user and computer object, etc.
4. At this point you’ve given the wizard all the information it needs to create a basic report. So you can place a check mark next to **Skip to the final page of this wizard without collecting additional data**, or click **Next** to fine-tune the results of the report.
5. On the **Advanced Simulation Options** page, you can choose to modify the Group Policy behavior in any of the following ways:
 - Simulate a slow link.
 - Simulate loopback processing, using either the **Replace** or **Merge** setting. (We’ll talk more about loopback processing in a minute.)
 - Specify which site to process, if you have GPOs attached to your Active Directory sites, and then click **Next**.
 - Specify which user and computer security groups you want to analyze. You can use security groups to do advanced filtering of GPO settings, which we’ll talk about in the “Applying Security Filtering” section later in the chapter.
 - Specify which user and computer WMI filters you’d like to simulate.
 - At this point you’ll be taken to a final screen that will list all of the settings you’ve selected. As with most wizards, you can click **Back** to make any changes, or click **Next** and then **Finish** to run the Modeling report.

Once you've completed the wizard, GPMC will create its now-familiar HTML report that will detail which GPO settings would be in effect in the situation you created. Perhaps most useful is that, if there are multiple Group Policy Objects present, the Modeling report will inform you which GPO "won." This is invaluable as an aid to troubleshooting, especially if your GPO structure is a complex one with multiple levels of inheritance.

Monitoring Group Policy Results

The Group Policy Results Wizard is quite similar to Group Policy Modeling, except that it provides the *actual* GPO settings that are being applied to a specific user/computer combination. Like the Modeling Wizard, it creates an HTML report detailing the GPO settings in place, and which GPO is enforcing those settings.

Caution You can't run the Group Policy Results Wizard for computers running Windows 2000. However, you can use Group Policy Modeling to basically mimic the same report. Yeah, I don't get it either; but there it is.

Who Gets What? Deploying Group Policies

Once you've customized your Group Policy Objects, you need to incorporate them into Active Directory so that your users can receive the appropriate settings. You accomplish this by *linking* Group Policy Objects to various containers within Active Directory: sites, domains, and Organizational Units. Once you've linked a GPO to a container, every object within that container will receive the GPO settings by default. In this section we'll look at how to link GPOs throughout your AD infrastructure, and how GPOs interact with one another if you have multiple objects linked to different points in your AD tree. We'll close with a look at some more advanced deployment topics such as controlling GPO inheritance and using security groups to fine-tune GPO deployment.

Using Organizational Units

The most common scenario for Group Policy deployment is to place the users and computers that require similar settings into a separate Organizational Unit, and then link a GPO to the OU to create a consistent configuration for all the members of that OU. You can link a GPO at the same time that you create it, or create an *unlinked* GPO and manually create a link once you've tested and finalized all of its settings.

- To create and link a GPO to an OU, open the Group Policy Management Console. Right-click the OU and select **Create and Link a GPO Here**.
- To link an existing GPO to an OU, open the GPMC. Right-click the OU and select **Link an Existing GPO**.

Note You can link a single GPO to multiple sites, domains, or OUs within Active Directory. However, it's a best practice not to link a GPO from one domain to a container in another, since this can create performance issues.

Configuring Policy Inheritance

In a complex network, you may find yourself with numerous GPOs deployed at various points throughout the infrastructure. It's important to understand how these different policies will interact with each other and ultimately affect your users. Much like NTFS permissions in the Windows file structure, Group Policy settings adhere to specific rules of *inheritance*. When a user logs onto your network, policy objects will be applied in the following order:

1. First, the Local Group Policy Object will be applied.
2. Second are any GPOs applied to the site the user and computer belongs to.
3. This is followed by any GPOs applied to the user's domain.
4. Finally, any GPOs linked to Organizational Units will be applied. If you have a nested OU structure, the GPO linked to the topmost OU will be applied first, and then the GPOs of any child OUs.

GPO inheritance is a cumulative process: this means that settings applied by later GPOs will be added to any earlier settings, rather than overwriting any previous settings. So if the domain GPO sets a minimum password length of eight characters, and then an OU GPO is applied that mandates a uniform screen-saver setting, the user will receive *both* settings.

So what happens if these additive settings conflict with each other? Say the domain GPO in your domain has a linked policy that blocks access to Registry editing tools, but you have created a Development OU that houses programming staff who require access to the Windows Registry. You create a GPO and link it to the Development OU, and explicitly grant access to Registry editing tools. Will this work? It will, because Group Policy Objects that are applied later have precedence over those that are applied earlier. (Think of it as having an argument where the person who gets in the last word is the one who wins.) So in this case, the Registry editing tools setting that was applied by the Development GPO “wins” over the setting applied in the

domain GPO. So users in your Development OU will have access to the Registry editor, but other users in the domain (provided they don't have other GPOs applied elsewhere) will not be able to access them.

Caution These inheritance rules do not apply to those settings that can only be set at the domain level: account policies, account lockout policies, and Kerberos policies. Even if you set different values for these items at an OU level, your domain users will still be held to the settings configured for the domain.

Customizing Policy Inheritance

What we've just described is the default behavior of Group Policy inheritance. But like anything else within Active Directory, you can customize these rules to finely control the way that Group Policies are deployed throughout your network. You can do this using security filtering with Active Directory security groups and Windows Management Instrumentation WMI filters, and by changing the default inheritance behavior for certain Organizational Units within your AD structure.

Blocking GPO Inheritance

If you have an OU that requires a very specific configuration, you may decide that you only want a single GPO to apply to it, so as to avoid interactions with other policy objects. You can accomplish this by right-clicking the OU within the Group Policy Management Console and selecting **Block Inheritance**. As the name suggests, this will prevent any GPO settings elsewhere in Active Directory from being applied within this particular OU; it will only receive settings from GPOs linked directly to the OU itself.

But in this case, even the exceptions can have exceptions: you can right-click a particular GPO *link* (not the GPO object itself) within the GPMC and select the **Enforced** option. This will ensure that the settings applied by this particular link cannot be blocked by any containers further down the GPO processing line.

Note In Windows 2000, you'll enable the **No Override** option on the **Properties** sheet of the GPO itself.

As you can imagine, overuse of these two options can wreak havoc on your network and make troubleshooting quite a challenge. What happens if you have a GPO attached to a parent OU that has the **Enforced** option enabled, and

then a child OU with a *different* GPO that has the **Enforced** setting enabled? If there is a setting conflict, which “enforced” GPO will win? In this case, the rules of inheritance will be reversed, and the *first* GPO applied with the **Enforced** setting enabled will take precedence. Tough to follow? I certainly think so: try to keep things simple and avoid convoluted scenarios like this one wherever possible.

Applying Security Filtering

Just like files and folders stored on NTFS volumes, you can create Access Control Lists for Group Policy Objects to control which Windows users and groups have access to them. In order for a GPO to be applied to a user or computer object, that object needs to have the **Read** and **Apply Group Policy** NTFS permissions to the GPO object. If both of these permissions are not present, the user or computer will not apply the settings within that particular GPO; in effect, it won't exist as far as that user/computer is concerned. When you create a new GPO in Windows Server 2003, for example, the following permissions are created by default:

- *Authenticated Users*: **Read, Apply Group Policy**
- *Enterprise Domain Controllers*: **Read**
- *Domain Admins/Enterprise Admins/SYSTEM*: **Read, Write, Create All Child Objects, Delete All Child Objects**

You can modify these permissions to ensure that specific GPOs will be applied to certain users or groups within a site, a domain, or an OU, and likewise prevent them from being applied to others. Notice that Domain Admins and Enterprise Admins do not receive the **Apply Group Policy** permission by default. This is to ensure that administrators do not become “locked out” of operating system functions when they need to perform troubleshooting or make modifications.

A scenario that is particularly conducive to using security filtering is software installation, since you may have one application that needs to be installed for specific users located across several Organizational Units, but not for your entire domain or even the entire population of an OU. Rather than creating multiple GPOs for each OU that requires the software, or managing multiple links to a single GPO, you can deploy the software within the domain GPO, and then use security filtering to specify which users and groups should receive the software. Depending on your specific needs, you can adopt one of two strategies:

- Remove the **Read/Apply Group Policy** entry for the Authenticated Users group. Then manually add the specific users and groups who should receive the GPO settings, and grant each one **Read** and **Apply Group Policy** permissions.

- Leave the default permissions for Authenticated Users in place, and then explicitly deny the **Read/Apply Group Policy** permissions to groups who should *not* receive the GPO. Setting explicit **Deny** permissions to a specific group will override the default Authenticated Users permission, and those specific groups will not receive the GPO settings.

Either of these procedures will be effective, you simply need to decide which is more appropriate for your organization. As a general rule, adopt the strategy that will result in the most straightforward permission assignment for the GPO, since this will simplify any changes you need to make later, as well as any troubleshooting you need to do.

Using WMI Filtering

For Windows XP and Windows Server 2003 machines, you can also control which machines receive particular GPO settings through the use of a WMI filter. WMI allows you to query a computer about its specific hardware and software settings, such as which service pack is installed, how much free space is available on the C:\ drive, and whether a specific service is installed or running. You can then create filters, which consist of one or more queries based on this type of data, to control whether or not the GPO gets applied. If the result of the WMI filter (such as “Is Service Pack 2 installed?” or “Does the machine have 350MB free drive space?”) is true, then the GPO is applied to that destination computer. If not, then the GPO is ignored.

Caution On a machine running Windows 2000, any WMI filters will be ignored and the GPO will always be applied.

To create a WMI filter, follow these steps:

1. Open the GPMC console tree, right-click the WMI Filters node, and click **New**.
2. Enter a name and description for the WMI filter, and then click **Add**.
3. In the WMI Query dialog box, enter the text of the query that you want to run, such as

```
Select * from Win32_OperatingSystem where  
Caption = " Microsoft Windows XP Professional"
```

4. Click **OK** after you've entered your query. If you want to filter on more than one query, you can click **Add** to enter additional query information. When you're finished, click **Save**.
5. Select the GPO that you want to link this WMI filter to, and select the name of the filter in the WMI Filtering drop-down box near the bottom of the right-hand pane of the GPMC console.

Just as you can link a single GPO to multiple sites, domains, and OUs, so you can link a single WMI filter to multiple Group Policy Objects. However, you can only link a single WMI filter to a GPO at any one time. This is because processing a WMI filter, especially a complicated one involving multiple queries, can be resource-intensive for the target computer and can increase a user's logon time.

Controlling the Desktop

Perhaps the most visible effect of Group Policy Object is your ability to control the desktop environment for your end users' workstations. It can, however, create one of those tricky balancing acts between being in control of your desktops and being a control *freak* about them. Now, from an administrative standpoint, we all know that consistency is good: if everyone's desktop is configured the same way, it makes it that much easier to troubleshoot file incompatibilities or to make large-scale changes as applications need installing or upgrading. On the other hand, there's also a school of thought that says that too rigid of an environment makes for unhappy users (which can thus make for unhappy administrators). In most corporate environments, for example, you're not going to want to allow people to install games or other personal software on their business workstations, but is there really any harm in allowing some flexibility to customize their wallpaper, their screensaver, and the like? There *are* situations where such tight control is warranted, of course, such as a public kiosk in a library or an airport, or in a 24×7 customer service center where users share workstations over multiple shifts. Like anything else, it's a compromise; you need to decide where your organization's computers need to live on the "lockdown scale." Luckily, Group Policy allows you to grant varying levels of autonomy to different groups of users and computers, as we'll discuss in this section.

Configuring Lockdown (Kiosk) Workstations

In some cases, you'll want to lock down a workstation as much as possible, especially if it's in a public area like a library or a retail store. In many cases, such a machine will be used solely for accessing a specific web page to look up prices, check reservations, and the like. When configuring a GPO for this type of machine, you want to be as stringent as possible in controlling what the user is able to access and change. Some of the settings that you might want to enable (either from the GPMC or the **Group Policy** tab in Active Directory Users & Computers) include the following:

- Computer Configuration\Windows Components\Internet Explorer:
 - **Disable Automatic Install of Internet Explorer components**—Enabled
 - **Disable Periodic Check for Internet Explorer software updates**—Enabled
 - **Security Zones: Do not allow users to add/delete sites**—Enabled
 - **Security Zones: Do not allow users to change policies**—Enabled
 - **Security Zones: Use only machine settings**—Enabled
- Computer Configuration\Windows Components\Control Panel\Display:
 - **Hide Appearance and Themes tab**—Enabled
 - **Hide Desktop tab**—Enabled
 - **Hide Screen Saver tab**—Enabled
 - **Hide Settings tab**—Enabled
 - **Prevent changing wallpaper**—Enabled
 - **Remove display in Control Panel**—Enabled
- Computer Configuration\Windows Components\Desktop:
 - **Do not add shares of recently opened documents to My Network Places**—Enabled
 - **Don't save settings at exit**—Enabled
 - **Hide and disable all items on the desktop**—Disabled
 - **Hide Internet Explorer icon on desktop**—Enabled
 - **Hide My Network Places icon on desktop**—Enabled

- **Prevent adding, dragging, dropping, and closing the Taskbar's toolbars**—Enabled
- **Prohibit adjusting desktop toolbars**—Enabled
- **Prohibit user from changing My Documents path**—Enabled
- **Remove My Computer icon on the desktop**—Enabled
- **Remove My Documents icon on the desktop**—Enabled
- **Remove Recycle Bin icon from desktop**—Enabled
- **Remove the Desktop Cleanup Wizard**—Enabled

As you can see, these settings are quite rigid and designed for machines that are installed in public areas. You can certainly modify these settings to create a more relaxed desktop environment for machines that are “owned” by one particular individual. But the most powerful lockdown mechanism that you can (and should) employ involves controlling the applications that a user can launch from her client workstation; we’ll discuss this in detail in the next section.

Using Software Restriction Policies

High on the wish list of most administrators is the ability to restrict what kind of software can run on the workstations on their network. Windows NT and 2000 offered a certain amount of control in this area, but it ranged from “hit-or-miss” to “darn near impossible to configure.” You could disallow **freecell.exe**, for example, and a savvy client could simply rename the file to **notepad.exe** or another application on the permitted list. Once he did that, the blocked application would open as if you’d put no restrictions in place at all.

Windows Server 2003 has made significant advances in this area, providing nearly foolproof options for controlling how software is used on your network. This can be useful not only for restricting the use of games and other nonbusiness software on your client workstations, but also as a way to restrict viruses and malware. How is this possible? Imagine a virus that executes a VBScript to launch a Denial of Service attack. If you’ve configured software restrictions so that no VBScripts can run on your network, then the virus will be stopped even if someone accidentally opens an infected e-mail attachment. And even non-malicious software can create issues for an enterprise network if it hasn’t been tested and approved: system files can be overwritten and can create the dreaded DLL Hell that makes Windows troubleshooting such a joy.

SOAPBOX: BLAME THE USER?

Wouldn't our lives all get a whole lot simpler if we could make those pesky users go away and stop bothering us? I mean, they're *horrible!* Always clicking on attachments and needing more disk space and generally making a mess of our nice, orderly networks!

But despite our occasional frustrations, part of what separates good network admins from great ones is the ability to secure a network without driving their clients to open revolt. It's very easy to say "We wouldn't need antivirus scanners if people would just stop clicking on things they're not supposed to." But that's also a bit of an oversimplification, since it assumes that every client on your network is just as technically savvy as you are. And this, as we all know, is hardly ever the case. If your network security strategy is "Get users to stop doing things they shouldn't," then it's a plan that's doomed to failure. And that's because it *only takes one*—one person who was in a rush, or forgot, or got fooled by a forged e-mail header, or any number of things that could happen to any of us. The reason I'm a big advocate of technologies like Group Policy and Software Restriction Policies is that they help to protect your clients from themselves. And if your clients are protected, then so, by extension, is your network.

Software Restriction Policies begin with one of two configurations whereby you decide how applications should be treated on your network, called **Unrestricted** and **Disallowed**. The difference between the two is pretty obvious: you need to make a choice between "Run everything except the stuff I tell you is bad" versus "Don't run *anything* except what I explicitly tell you is allowed." Once you've made this initial decision, there are four rules that Windows Server 2003 can use to restrict software usage on your network: Path, Hash, Certificate, and Zone.

Caution If you don't use a test environment for anything else, I *strongly* urge you to create one before you deploy Software Restriction Policies. Imagine a worst-case scenario: if you create a policy, set the default to **Disallowed**, and then don't specify any programs that are allowed to run, you've just created a policy that won't allow *anything* to run. You wouldn't even be able to log on to a workstation or server that's been configured this way. Even in less extreme situations, this is a powerful tool that warrants thorough testing before implementing it on a production network.

Creating a Software Restriction Policy

Rather than drown you in details, I'll first walk through creating a basic Software Restriction Policy using some default options. Once you've got the big picture at that point, you can get into the nitty-gritty of each rule type

and some of the other advanced options that you can set within the policy. To create a Software Restriction Policy, follow these steps:

1. Open the target GPO in the Group Policy Management Console. (Right-click the object and click **Edit**.)
2. Navigate to **User Configuration > Windows Settings > Security Settings > Software Restriction Policies**.
3. Right-click the Software Restriction Policies folder and select **New Software Restriction Policy**.
4. Your first step is to decide whether your overall software policy will be **Unrestricted** or **Disallowed**. By default, a new policy will use the **Unrestricted** setting. To change this, right-click **Disallowed** and select **Set as default**. (But you really should configure rules for what programs *are* allowed to run before you do this.) For our purposes, we'll assume that you're leaving the default **Unrestricted** setting, and want to disallow specific programs instead.
5. Next, configure a Path rule to disallow a specific application. Let's say that you've been instructed to restrict use of the AOL Instant Messenger application on your network. Right-click the Additional Rules folder and select **New Path Rule**. You'll see the dialog box shown in Figure 4-4. Enter the path to the AIM executable, and set the security level to **Disallowed**. This change will take effect the next time that the GPO is refreshed, or when a user logs out and logs back in.

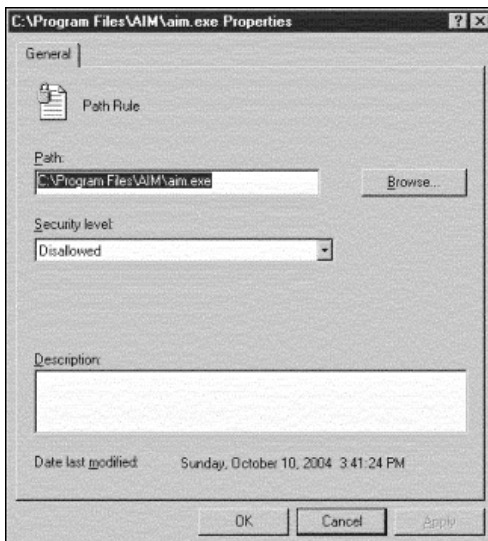


Figure 4-4. *Configuring a new Path rule*

■ Caution If you support Windows XP workstations, Software Restriction Policies will take two (I've even heard anecdotal reports of *three*) reboots to take effect. This is because of the Fast Logon Optimization feature, in which Windows XP doesn't wait for a network connection to come all the way up before applying Group Policies. Check out Knowledge Base Article 305293 for the full explanation, available at <http://support.microsoft.com/?id=305293>.

Configuring Zone Rules

A Zone rule allows you to restrict software based on the Internet Explorer zone that it was downloaded from: Internet, Local Intranet, Restricted Sites, Trusted Sites, or Local Computer. This would be useful if you use an intranet site to make software applications available to your users: they could install and run software downloaded from the Local Intranet zone, but not anything they downloaded from an untrusted game server.

■ Note Why are you using an intranet site to deploy software? Group Policy can do that for you! Never mind, we'll get there in a minute.

Before you break into a happy dance over how cool this feature is, though, you can only use it to regulate MSI installers, not .EXE files or other downloadable files. This makes it perhaps the *least* useful of the software restriction rules, unfortunately. Maybe it'll be improved in a future Group Policy or Windows operating system release, because it really is a great idea.

Configuring Hash Rules

One of the largest challenges of software restriction in Windows NT and Windows 2000 was the fact that restrictions were keyed to the file name of the executable that you were trying to allow or disallow. So you could disallow **sol.exe** or **kazaa.exe** all you wanted to, but all a crafty user needed to do was rename the executable to **notepad.exe** or a similarly innocuous name, and any software restrictions would be circumvented.

In Windows Server 2003, you can use a Hash rule to increase the effectiveness of your software restrictions. A *hash* refers to a kind of mathematical fingerprint that will uniquely identify a file regardless of where it lives within

the file system, and even regardless of whether it's been renamed. This fingerprint remains unchanged when the file is copied, moved, or even renamed. This means that our intrepid user's "rename the file" strategy would be foiled if Hash rules were in effect. Creating a Hash rule is nearly identical to the way we created the Path rule, except that you'll select only the file name rather than the full path.

Note Most antivirus companies will make the hash values of known virus files available to the public. You can then paste this hash into a Software Restriction rule to prevent the virus from running on your network.

Another great use for Hash rules is to prevent damage caused by viruses and Trojans that attempt to overwrite operating system files with malicious copies. So if your policy were configured to only allow the applications that you name, even if a virus could overwrite **WINWORD.EXE** with a malicious Trojan, it still wouldn't be able to launch because the hash value would not match the one specified in the Hash rule.

Configuring Certificate Path Rules

You can also configure Software Restriction Policies to use certificates to determine whether software can run or not. For example, you can use Certificate rules to automatically trust software from a third-party vendor or from within your organization. Certificates used in a Certificate rule can come from a commercial CA like Thawte or VeriSign, a Windows 2000/Windows Server 2003 PKI server, or a self-signed certificate. This is a really useful way to prevent users from downloading unauthorized ActiveX controls from untrusted websites.

Configuring Path Rules

As you saw in the sample policy we created, Path rules can specify the fully qualified path to a program. You can also use wildcards and folder names to create less-specific Path rules. When a Path rule specifies a folder, it will apply to any program that's contained in that folder, as well as any programs contained within any subfolders. You can use both local and UNC paths to create a Path rule, as well as environmental path variables like %WINDIR%.

Caution Use system variables with caution since they are client-specific. If a user can modify her local environment variables, it can affect the results of any Path rules that rely on those variables.

You can also use the familiar `*` and `?` wildcards to increase the flexibility and effectiveness of your Path rules. For example, `*\Windows` will apply to `C:\Windows`, `D:\Windows`, and `E:\Windows`, in case your clients have their OS installed to a nondefault logical drive. You can also use wildcards for such familiar tricks as `*.exe`, `*.vbs`, and the like.

If you need even more flexibility than wildcards offer, you can control your Path rules using Registry paths. This is especially useful if you need to restrict the contents of an application that may not be installed in a consistent location, but that stores its installation directory within a Registry key. So a Registry Path rule could look up the value in a Registry key such as this:

```
%HKEY_LOCAL_MACHINE\SOFTWARE\VendorName\AppName\  
Directories\Install Dir%.
```

When creating a Registry Path rule, you'll use the following format:

```
%[Registry Hive]\[Registry Key Name]\[Value Name]%
```

Path Rule Precedence

There's a specific order in which multiple Path rules will be enforced, depending on how specific the policy is. What does this mean? Essentially, a Path rule that is defined on a specific file (a more restrictive rule) will take precedence over policies applied to a file folder, or to policies involving wildcards (less restrictive rules). Any conflicts between Path rules will be resolved using the following precedence:

1. *Drive:\Folder1* will be applied first and has the lowest precedence.
2. *Drive:\Folder1\Folder2*.
3. **.Extension*.
4. *Drive:\Folder1\Folder2*.Extension*.
5. *Drive:\Folder1\Folder2\FileName.Extension* will be applied last and has the highest precedence.

Software Restriction Rule Precedence

In addition to resolving conflicts between Path rules, you'll need to understand how the different restriction types interact with each other as well. If multiple rule types are in effect, policies will be applied in the following order:

1. The Internet Zone rule has the lowest precedence of all Software Restriction Policies.
2. Path rules, in the following order:
 - *Drive:\Folder1*
 - *Drive:\Folder1\Folder2*
 - **.Extension*
 - *Drive:\Folder1\Folder2*.Extension*
 - *Drive:\Folder1\Folder2\FileName.Extension*
3. The Certificate rule.
4. The Hash rule has the highest precedence of all Software Restriction Policies, and will be applied last so that its settings “win.”

Much like Group Policies, the last policy that applies is the one that takes precedence. So if you create a Hash rule that allows **Unrestricted** access to **iexplorer.exe**, but define a Path rule that disallows it, the program will be allowed to run.

If after all of this you *still* have two identical rules that are applying differing security levels to the same executable, the more stringent rule will take precedence. For example, if two Hash rules—one with a security level of **Disallowed** and one with a security level of **Unrestricted**—are applied to the same software program, the **Disallowed** rule will take effect and the program will not run.

Note As with any configuration policies on your network, I'm going to tell you that your best bet is to keep it simple. Applying multiple policies and worrying about precedence rules is mostly going to add to troubleshooting difficulties and not much else.

Securing Client Operating Systems

Another useful item in the Group Policy bag of tricks is that you can use it to create a standard security configuration across your entire network, without needing to visit individual machines to repeatedly perform the same configuration. (We all know how tedious that is, not to mention error-prone.) By using *security templates*, you can create a security policy for your entire network or for a smaller group of similarly configured computers. You can do this using one of the predefined security templates included in the Windows 2000 or 2003 OS, or you can modify one of these templates or even create a brand new one containing the specific security settings that you need. Security templates can be used to define the following components:

- Account policies
- Password policies
- Account lockout policies
- Kerberos policies
- Local policies
- Audit policies
- User rights assignments
- Security options
- Event log settings
- Restricted groups settings
- System services: startup modes and permissions
- Registry key permissions
- File and folder permissions

In this section, we'll look at the steps for implementing security templates on an Active Directory network. The process begins with analyzing the current security settings on various machines on your network, creating or modifying a security template containing your desired settings, and then importing that template into Group Policy so that it can be rolled out to your entire network.

Analyzing Current Security

You can analyze the security settings on any machine in one of two ways: either by using the MMC, or through the **secedit** command-line utility. The Security Configuration and Analysis console (which I'll refer to as SCA from this point on) isn't one of the default consoles installed in your Administrative

Tools folder; you'll need to open a blank MMC console in author mode (enter **mmc /a** from the Run line) and use the **File ► Add/Remove Snap-in** menu command.

Note Since you'll probably be using this tool fairly often, I'd recommend that you save a custom console to your Administrative Tools folder for easy access.

Using SCA is pretty intuitive since the opening screen provides you with instructions to get started. In order to work with SCA, you'll need to create a database that will store the security template values that you're comparing or applying. If you have an existing database, simply right-click the SCA node and select **Open**, and then browse to the database file you need. If you're creating a new database, you'll need to do the following:

1. Right-click the SCA node and select **Open Database**. (I know it seems counterintuitive to open a database that doesn't exist yet. Trust me, it works.)
2. Enter the path and name of the SCA database that you want to create, and then click **Open**.
3. You'll then need to select the template that you want to compare your current settings against. We'll go into detail about what the different templates do in a later section, but for now the file names should look fairly intuitive:
 - *compatws.inf*: Used for workstations that need backwards compatibility with legacy applications or networks.
 - *dcsecurity.inf*: This is created by the operating system when a member server is promoted to domain controller status.
 - *iesacsl.inf*: Allows you to lock down Internet Explorer settings.
 - *hisecc*.inf*: Used for a high-security configuration; *hiseccdc.inf* corresponds to a domain controller, *hiseccws.inf* is for a secure workstation.
 - *notssid.inf*: Used to remove the Terminal Server user SID from a server that isn't being used for Terminal Services connections.
 - *rootsec.inf*, *setup security.inf*: Ignore these for now, we'll discuss them in the "Using Security Templates" section in a moment.
 - *secure*.inf*: Used for situations where you want a secure configuration, but the settings in the *hisecc*.inf* templates are a bit over-the-top. Sufficient for most environments.

4. Once you've selected a template, you'll then *analyze* your computer's security settings compared to those within the template. (The instructions that you see in the SCA GUI at this point describe the steps to *configure* your computer before the steps needed to analyze it: please, oh please, don't take this literally. You always want to analyze a computer's settings before blindly applying a new template.)
5. Right-click the SCA node again and select **Analyze Computer Now**. You'll be prompted for a location to store the analysis results as a .LOG file. Select a location, and then click **OK** to begin the analysis.

Once you've analyzed your computer's settings, you can browse through the SCA console to see where your settings differ from those within the template. You'll see a screen similar to Figure 4-5 with three columns: Policy, Database Setting, and Computer Setting. A red X will be displayed if a defined setting doesn't match the setting specified in the template, versus a green check mark if your settings are consistent.

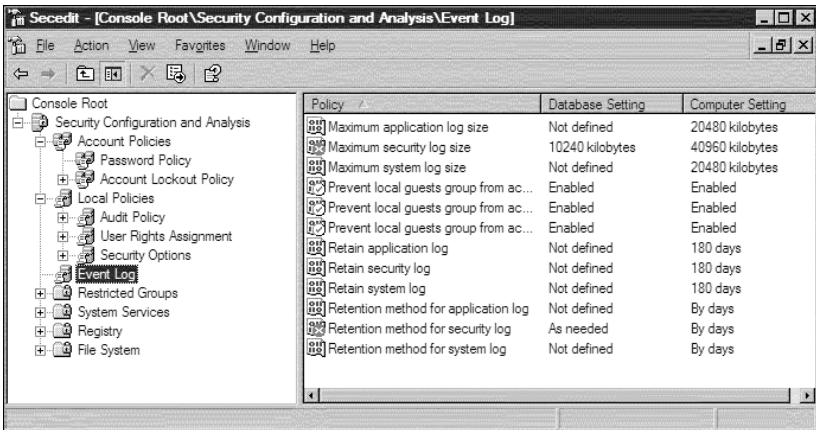


Figure 4-5. Browsing the *secedit* database

You can also perform a security analysis from the command line, using the *secedit* command with the following syntax:

```
secedit /analyze /db FileName.sdb [/cfg FileName] [/overwrite]
      [/log FileName] [/quiet]
```

Command-line arguments:

/db FileName specifies the database file used to perform the analysis

`/cfg FileName` specifies the security template to import into the database prior to performing the analysis. Security templates are created using the Security Templates snap-in.

`/log FileName` Specifies a file in which to log the status of the configuration process. If not specified, configuration data is logged in `Scesrv.log`, which is located in the `%windir%\Security\Logs` folder.

`/quiet` Specifies that the analysis process should take place without further comments.

Note The `/quiet` option comes in quite handy if you want to perform a security analysis on all of the workstations in your network: simply add a command like `secedit /analyze /db hisecws.sdb` to a batch file or login script, and you can collect information from your client workstations with very little effort.

Using Security Templates

So you've already seen a glimpse of security templates in action, but what are they really about? Just as a template for a word processor can help you create a document according to certain standards, a security template is simply a premade file that simplifies the process of comparing your computer's existing security settings against a predefined list. There are dozens (if not hundreds) of options available to you when securing a server, and the number of possible combinations is enough to make even a security expert's head spin. Templates allow you to quickly create a secure baseline for your network clients and servers and make it easier to control how configuration changes occur.

Windows 2000 and 2003 both come with a number of default security templates installed in the `%SystemRoot%\Security\Templates` folder. Nine default templates are installed with the operating system, and numerous others can be downloaded from the Microsoft website. Each of the default templates has specific characteristics, described as follows:

- *compatws.inf*: Provides the ability for members of the local Users group to run software that typically requires Power User or local Administrator access. This relaxes the permissions that are normally assigned to the Users group so that you don't need to add your users to these more powerful security groups.

- *DC security.inf*: This template is applied when a 2003 server is promoted to domain controller status. The template contains modifications specific to domain controller security, including file system rights and registry permissions.
- *securedc.inf*: Designed to increase security for domain controllers, including settings for passwords, account lockout, and auditing policies. This template also increases restrictions for anonymous users.
- *securews.inf*: Similar to the settings in *securedc.inf*, but designed for workstations and member servers.
- *hiseccd.inf*: Provides additional security (over and above that provided by the *securedc.inf* template) for domain controllers.
- *hisecws.inf*: Increases security for member servers and workstations.
- *iesacsl.inf*: Increases the default security configuration for Internet Explorer.
- *notssid.inf*: By default, Windows 2000 and 2003 add file system permissions and user rights for Terminal Services users on each server. If you know for certain that the server you're installing will not be used as a Terminal Server, you can apply this template to remove the unnecessary entries.
- *rootsec.inf*: Defines permissions for the root of the system drive; you can use it to reapply the root directory permissions if they are inadvertently changed. You can also modify this template to apply a specific set of permissions to the root of different volumes. This template doesn't overwrite any permissions that you've explicitly defined on any child objects below the system root; it merely propagates the permissions that are configured to be inherited by child objects.
- *Setup security.inf*: This template is actually created individually whenever you install a new Windows 2000 or 2003 computer; it allows you to revert the configuration of the machine back to its original settings at any time. This should obviously be used with extreme caution, since you'll be overwriting any configuration changes that you've made since the machine was initially installed.

Creating or Customizing a Security Template

If one of these preexisting templates meets your needs, you can apply it directly to your servers and workstations. You can also modify some of the settings to customize the template to address the specific requirements of your organization; however, it's a good idea to make a *copy* of the default template before making any changes to it. This way if you make a mistake or change your mind, it's easy to roll back to the default settings. To create a copy of a default template, do the following:

1. Open an MMC console in Author Mode (**mmc /a** from the Run line), and add the Security Templates snap-in.
2. Drill down to **Security Templates > C:\Windows\security\templates**.
3. Right-click the template you want to copy, click **Save As**, and enter a new name for the copy.

Note To create a blank template from scratch, right-click the C:\Windows\security\templates node and select **New Template**.

Once you have the new template in place, you can manually edit its settings by browsing through the various nodes in the Security Templates console. Alternatively, you can copy a specific group of settings from one template to another. For example, to copy the account policies settings from the hisecws.inf template into a blank one, follow these steps:

1. Navigate to **C:\Windows\security\templates > hisecws.inf**.
2. Right-click the Account Policies node and select **Copy**.
3. Navigate to the Account Policies node in your new policy, right-click, and select **Paste**. This will add the account policies from the hisecws.inf template while leaving the other nodes undefined.

Importing a Template into Group Policy

Once you've created a template containing the security settings you need, you'll import the template's settings into a Group Policy Object in order to propagate those changes to the machines on your network.

UNDERSTANDING DOMAIN-LEVEL POLICIES

It's important to keep in mind that certain Windows security policies can be set only at the domain level. These include the settings in the Account Policies node: account policies, account lockout policies, and Kerberos policies. This means that an Active Directory domain can have only *one* of these policies in effect at any given time: all users within a single domain will be bound to a single policy for things like password length and complexity, frequency of password changes, PKI policies, and Kerberos settings. The only exception to this is if you create a separate account policy on an Organizational Unit (OU) containing member servers. In this case, the *local* user accounts on machines within a given OU can have a different account policy apply to them. However, any *domain* accounts, even within a separate OU, will adhere to the domain account policy. If you have a significant portion of your user base that requires very different policies for account passwords, lockouts, etc., then you should consider creating a separate domain. Because of the transitive trusts created by Windows 2000 and Windows Server 2003, managing multiple domains isn't nearly as tedious as it was under Windows NT. However, maintaining separate domains will still add a level of complexity to your Active Directory environment; be sure when planning your AD infrastructure that you carefully consider these domain-level policies before creating an unworkable Active Directory structure.

To import a security template into a Group Policy, you'll need to do the following:

1. Open the GPO you wish to edit from the Group Policy MMC or the GPMC console.
2. Navigate to **Computer Configuration ► Windows Settings ► Security Settings**.
3. Right-click the Security Settings node and select **Import Policy**. You'll be prompted to browse to the .INF file that you want to import.

Caution A bit of Group Policy weirdness: importing a security template does not register as a “change” to a GPO. This means that the new settings won't be detected by your clients or servers when they query Active Directory for any changes to the GPOs. In order to resolve this, you should manually change a setting within the GPO, even if it's one you intend to change back later. This will ensure that your new security settings will be transmitted to your network machines in a timely fashion.

Configuring Software Deployment

You can also use Group Policy to deploy line-of-business applications throughout your Active Directory network. This installation can take place silently, without the need for user intervention or assigning elevated privileges to your users at the desktop level. Software that's installed via Group Policy is *self-healing*, which means that any application files that become corrupted or deleted will be replaced automatically by the Group Policy Object. Depending on the needs of your environment, Group Policy software deployment can allow a user's applications to follow him no matter where he logs on to the network from, or ensure that a specific set of tools is available on a particular machine no matter who logs on to it. In this section, we'll look at some of the most useful options available to you in using Group Policy to deploy software.

Creating an Installation Package

As long as you have an .MSI installer for the application you want to deploy, doing so through Group Policy is pretty much a snap. If your application does not have an .MSI file associated with it, though, you are still not entirely out of luck. You can create a .ZAP file that will still allow you to deploy the software, with the following caveats:

- The installation process can't take advantage of elevated privileges for installation. So if your users are only members of the Users group and they need Administrator access to perform the installation, the deployment will fail.
- The program can't be installed on the first use of the software—we'll talk about how .MSI does this in a moment.
- You won't be able to install a feature on the first use of the feature, similar to how Microsoft Word can leave the Thesaurus function uninstalled, but you can copy it to the user's workstation the first time she tries to use it.
- Most problematic of all, you can't roll back an unsuccessful installation, modification, repair, or removal of a .ZAP file the way you can with .MSI.

Note With more and more applications complying with the Microsoft Logo Program, this is a much smaller concern now than it was even when Windows 2000 was first released.

To create a software installation package for an .MSI installer, follow these steps:

1. Open the GPO that you want to use from the GPMC console.
2. Navigate to **User Configuration ► Software Settings ► Software Installation** from either the Computer Configuration or User Configuration node. (You can also deploy software to computers instead of users; we'll talk about that in the "Understanding Deployment Options" section next.)
3. Right-click the Software Installation node and select **New ► Package**. Browse to the location of the .MSI file and click **OK**.

Caution Since your network clients will need to access the .MSI file in order to perform the installation, be sure that it's located on a shared network drive and assigned the appropriate NTFS permissions.

4. The next screen gives you a choice of how you want to deploy the software: **Published**, **Assigned**, or **Advanced**. We'll go over the differences between these options next; for now select **Published**, which will install the application the first time a user clicks a file associated with it. (Double-clicking a .DOC file would launch the Microsoft Word installer, for example.)
5. Click **OK** to finish. The GPO Editor will take a moment to refresh itself, and then you'll see your software package listed in the Software Installation window. From here you can right-click the package and select **Properties** to change any deployment options.

Understanding Deployment Options

When deploying software, you need to make two major decisions:

- Do I want to publish this software package, or assign it?
- Do I want to deploy this software to a user object or a computer object?

In this section we'll look at the differences between these choices, as well as some more advanced options available for software deployments.

Publishing Applications

Publishing an application will make that application available to your users at their next login. Once you've published an application, a user can install or uninstall it by using the Add/Remove Programs applet in Control Panel.

The installer will also launch through *document invocation*, that is, when the user tries to view or edit a file that requires the published application to open. This is a good way to roll out applications that might not be used consistently across your network, since you won't be performing the actual installation unless (and until) the user actively requires the software. Using Group Policy will still ease the installation process for your users since they won't need to remember share names or instructions for manually installing software.

Note You can deploy *published* applications only to user objects, not computers. It makes a lot of sense since, after all, what are the odds that your workstation will decide of its own volition that it needs to install Microsoft Word one day?

You have a few additional options available to you when publishing a software package. When you right-click the package and go to **Properties**, you'll see the screen shown in Figure 4-6 by clicking the **Deployment** tab.

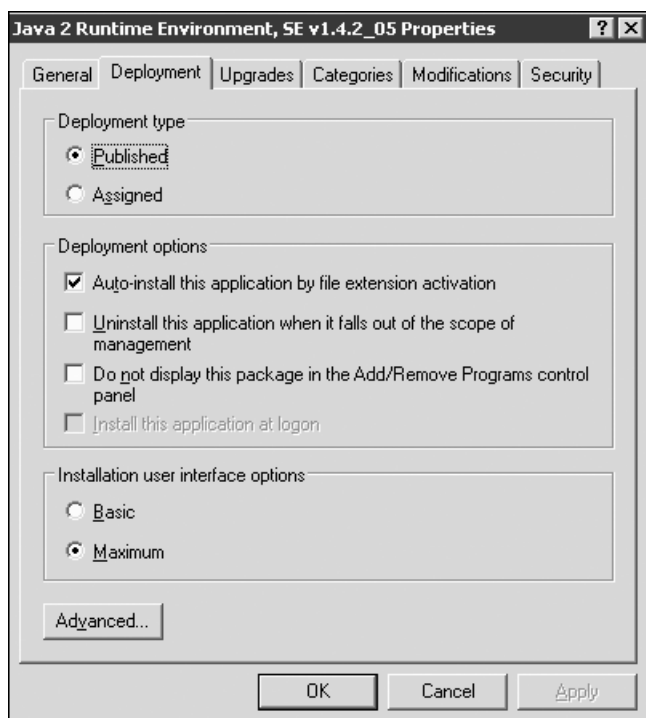


Figure 4-6. Configuring software deployment options

As you can see, the option to install the app when a user double-clicks the appropriate file extension is enabled by default. Two other options that you can enable are

- **Uninstall this application when it falls out of the scope of management:** Let's say that user JSmith is contained in the Accounting OU of your domain and has the PeachTree accounting package installed via Group Policy. If JSmith moves to Marketing, and the Marketing OU does not have the accounting software published to it, then the application will be uninstalled from JSmith's workstation. This is useful in ensuring that sensitive applications do not remain installed on a workstation if the user no longer has a need for them.
- **Do not display this package in the Add/Remove Programs control panel:** Just like it sounds, this ensures that a published application will *only* be installed through document invocation. You may enable this option to prevent applications from being installed unnecessarily by curious users.

Assigning Software

In addition to publishing an application, you can also *assign* it to either a computer or a user object. By assigning an application to a computer object, the application will be automatically installed the next time the computer boots up: this requires no document invocation or user intervention of any kind. Once the program has been installed, only an administrator will be able to uninstall it (either manually or through Group Policy). Like a published application, an assigned application is *self-healing* so that it can automatically repair or replace any damaged or erased program files.

Assigning an application to a user takes one of two forms. In the default scenario, the user will see a shortcut to the application on her Desktop or Start menu. However, the app won't actually be installed until the first time she double-clicks the shortcut or uses document invocation. And since the installation takes place silently, a user can easily be confused when he tries to launch the program and nothing seems to happen. It's important to be aware of this fact, since "I double-clicked the Excel icon and my machine has been hung up for like two minutes" can be a common help desk phone call in this situation.

While this was the only way of assigning software to a user in Windows 2000, Windows Server 2003 provides the **Install application at logon** option, which will perform an install as soon as the user logs on. Similar to the help desk calls you might experience from the default scenario, though, this option may greatly increase your users' logon times while the installation process is running. As with anything else, good communication with your users and support staff will help to make this operation as smooth as possible.

You'll typically assign software to computer objects for critical applications that need to be present on any computer on your network: antivirus software is a favorite use of this feature. Simply add the antivirus software's .MSI file to the Default Domain policy, and every machine in your network will receive the installation the next time they reboot.

Caution Installing applications with large source files can create congestion in your network traffic, especially if a large number of users request the installation at the same time. (At 9 a.m. when they arrive at the office, for example.) Be sure to take this into account when deciding which programs to assign to your users and computers.

Deploying Custom Applications and Upgrades

For applications with many different parts, such as Microsoft Office, you can even configure the installation file so that it only installs the components you want. The remaining components can be left out entirely, or you can allow them to be installed on their first use: the first time a user requests the Word spell-checker, for example. To customize your applications in this way, you'll use a *transform* file with the .MST extension. You'll specify these .MST files on the **Modifications** tab of the software package's **Properties** sheet, which you saw in Figure 4-6.

Finally, once you've deployed a software package through a GPO, you can use a newer installer to *upgrade* that package using the **Upgrades** tab of the **Properties** sheet. An upgrade package can either be optional or mandatory, and the upgrade will take place the next time the user logs on or the machine boots up.

Note Unlike other Group Policy settings that will refresh in the background every 90 minutes by default, software installation policies will only take effect at startup or logon. This is to prevent such catastrophes as a GPO trying to upgrade or uninstall a user's copy of Outlook while she's still trying to use it, for example.

Using Advanced Techniques

We'll close out the chapter with a few other Group Policy tricks that should be in any administrator's arsenal. This includes the ability to centrally configure permissions for all of your client workstations, as well as how to control membership to sensitive local and domain groups. At the end of the chapter

you'll find some links to online and print resources that I've found useful in creating Group Policy solutions for networks of all sizes.

Controlling the Registry and File System

One of the largest headaches for most network admins is the need to secure large numbers of client workstations in a quick and efficient manner. We've already seen how you can import security templates into Group Policy to deploy network-based security settings like minimum password lengths and account lockout policies, but you can also use a GPO to enforce security standards on your users' local hard drives. By browsing to **Computer Configuration ► Windows Settings ► Security Settings** within the GPMC, you can add entries to the following Group Policy nodes:

- System Services
- Registry
- File System

In the case of System Services, you can define how local services will behave on system startup, and which users and groups have permission to start, stop, or modify those services. If you remember the Code Red and Nimda worms, they attacked many workstations that had the IIS services installed. In many cases, the owners of these workstations didn't even know that their machines were running an instance of the IIS web server, and so were taken completely by surprise when these network attacks hit. You can use the System Services node to universally disable a service like World Wide Web Publishing, Telnet, or any other service that really shouldn't be running on a workstation. That way, even if a virus or spyware program attempts to start the service, the malicious software will be unable to do so.

The Registry and File System nodes allow you to set NTFS permissions on specific registry keys or file/folder paths. Simply add the full name of the Registry key or the folder path that you want to secure, and you'll see a familiar **Properties** sheet that will allow you to specify permissions just as though you were sitting at the console of the workstation itself. You'll also have the option to propagate the permissions to any subfolders or subkeys of the folder or key you specify.

Caution Note that none of these settings will *create* a service, Registry key, or file system path. These GPO settings are simply used to configure security on existing workstation configurations.

Using Restricted Groups

When you're protecting your domain and local user accounts, restricting membership to sensitive groups like Domain Admins, Enterprise Admins, and the like is absolutely critical. If malicious users, either external or internal, can somehow create an account for themselves that is a member of one of these groups, then the security of your entire Active Directory infrastructure can become irrevocably compromised. The solution to this is the use of restricted groups within Group Policy. By right-clicking the Computer Configuration\Windows Settings\Security Settings\Restricted Groups node and selecting **Add Group**, you can specify the following information:

- Which users or groups should belong to the restricted group, and
- Which users or groups should not belong to the group

Let's say you've restricted the Domain Admins group so that it can only contain the user accounts for yourself and two of your staff members. If you accidentally add (or delete) an account from Domain Admins membership, the Restricted Groups policy will re-create the membership list the next time that the policy is applied: every 90 minutes by default. You can also use this setting to restrict local group memberships on member servers and workstations.

Summary

In this chapter, we looked at the wide, wonderful world of Group Policies, and how they can make your life as an Active Directory administrator so much simpler by allowing you to specify security and usability settings for an entire site, OU, or domain from a single location. We started by looking at the Group Policy Management Console, which is a new Group Policy management tool that greatly simplifies the process of creating and managing Group Policy Objects. We then talked about the different configuration settings that you can control through Group Policy, including security settings like password policies and account lockout policies. You can also use a GPO to customize and control your users' desktops, removing access to potentially harmful items like the Control Panel and the command prompt. To further control security in your environment, you can use security templates that will allow you to configure numerous machines with the same security settings without risking data entry errors from entering the same information multiple times. You can also use Group Policy to centrally deploy software applications to your users and computers, as well as controlling the software that's allowed to run by using Software Restriction Policies. We closed out the chapter with a look at some more advanced Group Policy tricks, such as configuring permissions on Registry keys and NTFS files and folders, and controlling the startup behavior of Windows services.

Additional Resources

Windows Server 2003 Active Directory: <http://www.microsoft.com/windowsserver2003/technologies/directory/activedirectory/default.msp>.

Moskowitz, Inc., Group Policy Resource Center: <http://www.gpanswers.com>—Check out the forums and newsletter!

Group Policy, Profiles, and Intellimirror for Windows 2003, Windows 2000, and Windows XP by Jeremy Moskowitz. Sybex Publishing. ISBN 0-7821-4298-2—Order it from <http://www.gpanswers.com> and the author will even sign it for you.

Microsoft Technet Script Center: <http://www.microsoft.com/technet/scriptcenter/default.msp>—Great for the scripting or WMI beginner, including a repository of ready-made scripts and filters that you can use right away.

Active Directory Discussions Mailing List: <http://www.activedir.org>—Moderate traffic (50 messages in a day is pretty busy). Required reading for anyone who is serious about managing their Active Directory network well.