

If you plan to use security-group filtering on your GPOs to any large degree, I highly recommend that you use a tool that supports Resultant Set of Policy (RSoP) calculations, such as FAZAM 2000 from FullArmor Corp. (<http://www.fullarmor.com>). Such tools make troubleshooting these kinds of scenarios simpler by providing you with information about what the RSoP is for a particular user on a particular computer in a particular OU.

Using GPO Functionality

In this section, I'm going to drill into using some key aspects of GPOs. In particular, the Administrative Template policy, Software Installation, Folder Redirection, and Security Settings features warrant further discussion. Each policy section has its own quirks and configuration options. So it's useful to describe how you can use each one in situations that you're likely to encounter in your own environment.

Administrative Template Policy

Administrative Template policy was born out of NT 4.0 system policy. As discussed in Chapters 1 and 2, Administrative Template policy lets you set HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER Registry values using the familiar .adm template files, which are stored in the GPT for each GPO. In Chapter 1, I also mentioned the difference between preferences and policies as well as the benefits that policies provide by their lack of tattooing of the Registry. In this section, I'll walk through setting, and techniques for effectively editing, Administrative Template policy.

To begin with, let's set up some Administrative Template policy. I'm going to open a Group Policy snap-in focused on my Finance OU, called Finance GPO. Let's suppose I'm setting Administrative Template policy on a per-user basis. Expand User Configuration\Administrative Templates, as shown in Figure 3.17.

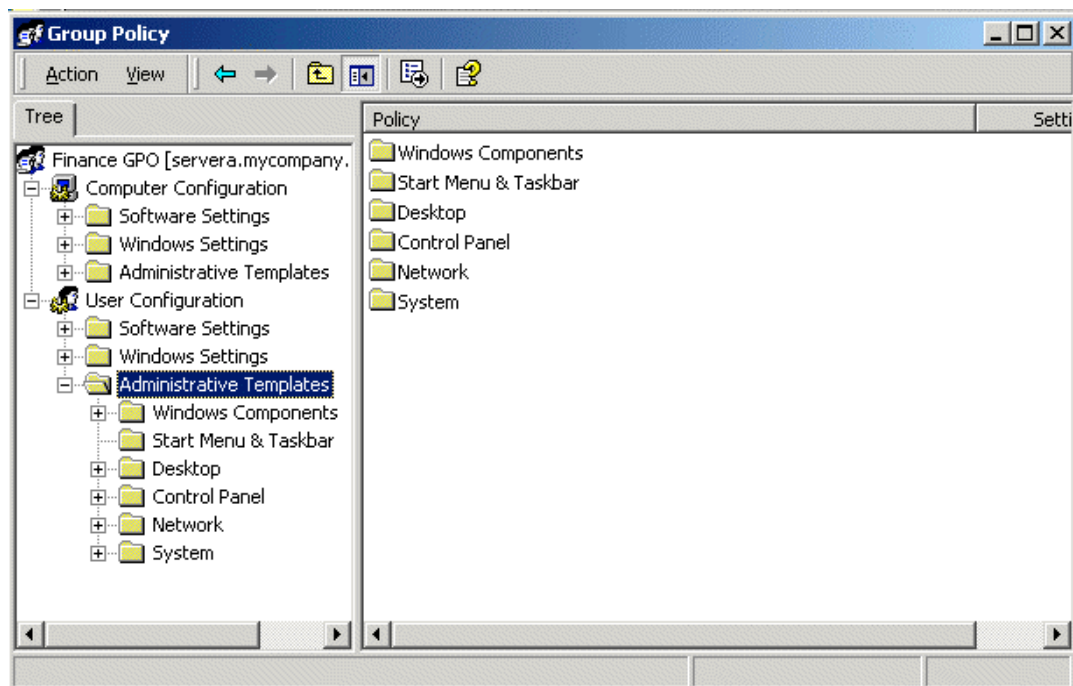


Figure 3.17: Viewing user-based Administrative Template policy.

In this section, I'm going to set some common desktop lockdown policies. To do this, I need to drill down into Windows Components\Windows Explorer. The policies available under this node include restrictions such as Remove "Map Network Drive", "Disconnect Network Drive," and Prevent Access to Drives from My Computer. To set a particular policy, I double-click that policy item in the right-hand pane of the Group Policy snap-in.

All Administrative Template policy takes a form very similar to that shown in Figure 3.18.

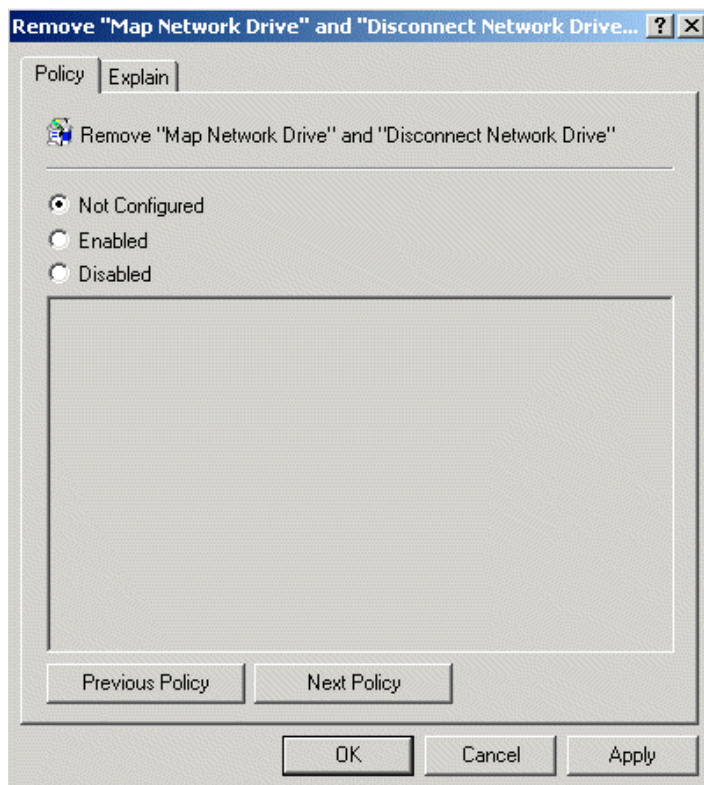


Figure 3.18: Viewing a typical policy setting in Administrative Templates policy.

This dialog box gives me three options for the policy—Not Configured, Enabled, and Disabled. As described in Chapter 1, these settings ignore, set, and de-select this policy item. This dialog box also contains an Explain tab. This is a new feature in Win2K Administrative Templates, and it provides a detailed explanation of what a policy setting does. The Explain text, as well as the Registry key and value behind the policy setting, are stored in the .adm file(s) that are part of this GPO's GPT, as discussed in Chapter 2.

After you've set a number of policy items, there's no easy way to see where all policy has been set in a GPO. To get around this, Microsoft provides the Show Configured Policies Only command on the Group Policy snap-in View menu. Choosing this command displays only those Administrative Template policies that have a value other than Not Configured. This is illustrated in Figure 3.19, where only policy defined in the Windows Components\Windows Explorer section has been set on this GPO.

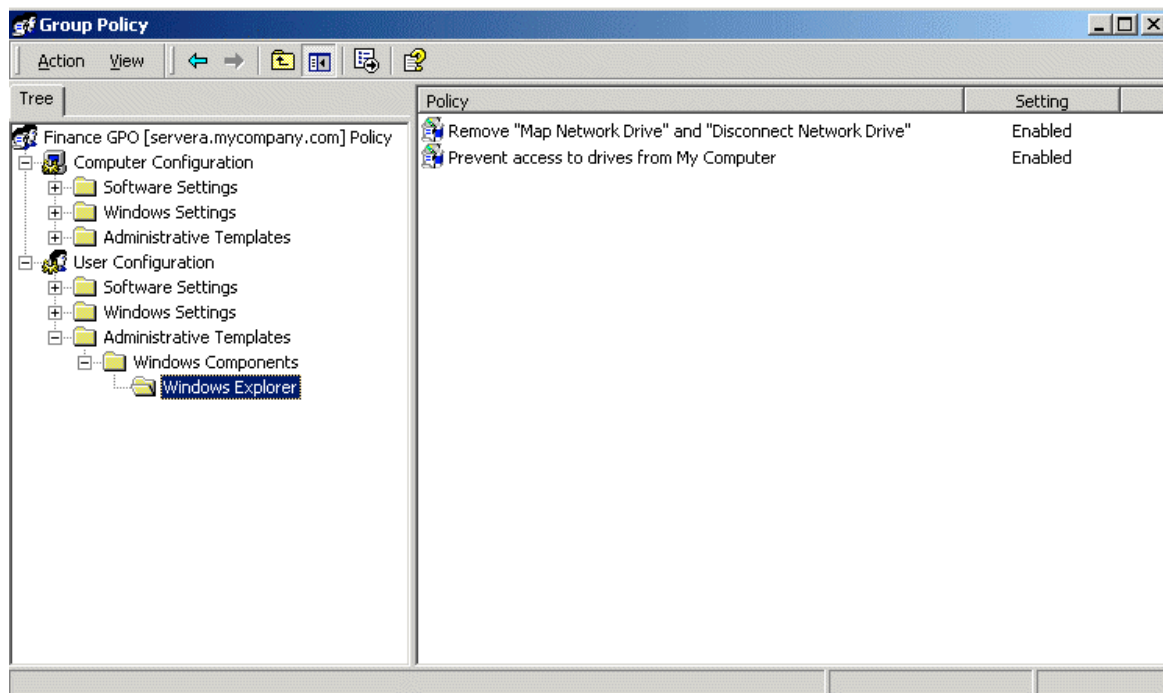


Figure 3.19: Viewing only Administrative Template policy that has been set.

If you've modified or defined your own custom .adm files, you must load them into the GPO using the Group Policy snap-in tool. Specifically, if you right-click the word *Administrative Templates* on either the User or the Computer configuration nodes, then choose Add/Remove Templates from the context menu, you'll be presented with the Add/Remove Templates dialog box, shown in Figure 3.20.

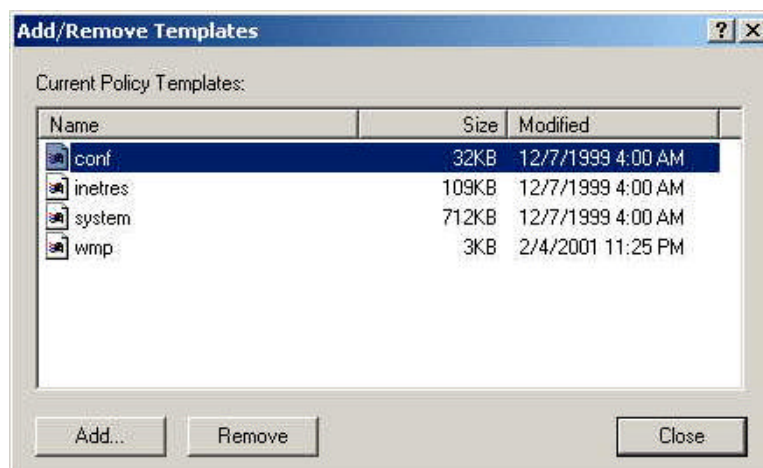


Figure 3.20: Viewing the Add/Remove Templates dialog box in Administrative Templates policy.

From here, you can add your new or modified .adm file to the GPO. Because .adm files are stored in individual GPOs, if you want the .adm file to be used across multiple GPOs, you'll need to repeat this loading operation on each GPO. (Or you can make the change on one GPO that you've designated to hold your .adm changes, then link to it from multiple containers.)

If you click Add in the Add/Remove Templates dialog box shown in Figure 3.20, the familiar Open dialog box appears, focused on the %systemroot%\inf folder on the computer where the GP snap-in tool is running. This folder typically contains the default .adm files shipped by Microsoft in Win2K. (If your .adm file is stored in a different folder, you can, of course, point to it.) When you select the appropriate .adm file and click Open, the file is automatically copied to the ADM folder in the GPT defined for that GPO (as described in Chapter 2). The .adm file is now part of the GPT, and the source file that you copied it from remains separate and independent.

If you have to make a change to the original .adm file, you need to edit the copy that is now in the GPT in the ADM folder. Or you must edit your local copy, remove the template file from the GPO, then add it again using the new version.

While it's impossible in the space of this book to define all of the policy items available in the default .adm files shipped for use by Administrative Template policy, Table 3.1 attempts to present the major categories of Administrative Template policy and their purposes. This table includes both policies and "preferences," as I defined them in Chapter

This Policy	Provides Policy For This
Computer & User Configuration Windows Components NetMeeting	Per-machine and per-user settings for the NetMeeting conferencing Application. For example, this policy allows you to disable the whiteboard or chat features in NetMeeting for users subject to this policy.
Computer & User Configuration Windows Components Internet Explorer	Per-machine and per-user settings for Internet Explorer. This policy includes items that let you lock down IE, such as preventing users from changing security-zone policy and accessing certain tabs on the IE Internet Options page. This policy section provides complementary IE controls to the Internet Explorer Maintenance policy node in a GPO.
Computer & User Configuration Windows Components Task Scheduler	Per-machine and per-user settings for the Task Scheduler. Both computer and user settings are identical, and they let you control things like users' ability to create scheduled tasks or delete existing tasks.
Computer & User Configuration Windows Components Windows Installer	Per-machine and per-user settings that let you control the behavior of the Windows Installer engine. On the per-machine side, you can set policy such as enabling Windows Installer logging or disabling the Installer engine altogether. On the per-user side, you can control behavior such as application rollback (when an installation fails) and whether the Installer engine can escalate a user's privilege level during installation.
User Configuration Windows Components Windows Explorer	Setting a number of Explorer user interface (UI) locks on a per-user basis only. For example, included in this policy section are lockdowns such as hiding particular drives in My Computer and removing the Map Network Drive and Disconnect Network Drive options from Explorer.



This Policy	Provides Policy For This
User Configuration Windows Components Microsoft Management Console	On a per-user basis, which MMC snap-ins users can load; also, whether users can enter MMC Author mode from a pre-defined MMC console tool.
Computer & User Configuration System	Per-machine and per-user settings for miscellaneous system lockdowns. For example, under the per-machine policy section, you can disable the CD-ROM autoplay feature and remove the Welcome screen at logon. Similarly, on the per-user side, you can disable Registry-editing tools and specify only a particular list of applications that a user can run.
Computer Configuration System Logon	Controlling behavior at logon, including whether logon and startup scripts run synchronously; also, what constitutes a slow network connection for roaming user profile download behavior.
User Configuration System Logon/Logoff	Similar restrictions to the previous per-machine logon policy. Also adds restrictions that let you remove certain options from the Start menu, like Log Off, and certain elements from the Ctrl-Alt-Del dialog box, such as the Task Manager.
Computer Configuration System Disk Quotas	Per-machine policy that lets you set disk quota behavior, such as whether to enable disk quotas, whether to enforce them, and how to warn the user if they're exceeded.
Computer Configuration System DNS Client	Per-machine policy that lets you specify a DNS-suffix search order on machines that are subject to this GPO. A suffix search order is used if you have DNS subdomains on your network that you want to search when trying to resolve unqualified host names.
Computer & User Configuration System Group Policy	Per-machine and per-user policy that lets you control the behavior of GPO processing. On a per-machine basis, this policy lets you specify whether a particular CSE can process GPOs in the background (when supported), how frequently background processing occurs, and whether to process a GPO if it hasn't changed. (The default behavior is to not process GPOs that haven't changed.) In addition, per-machine and per-user settings determine when a slow network connection is detected and how that affects GPO processing. In some cases, like Folder Redirection and Software Installation, those CSEs don't run when a slow network is detected.
Computer Configuration System Windows File Protection	Per-machine policy that lets you control the behavior of Windows File Protection (WFP). For example, you can control how often WFP scans critical files for changes and how big a cache should be allocated to store the correct



This Policy	Provides Policy For This
	versions of system files should the current copies get corrupted.
Computer & User Configuration Network Offline Files	Per-machine and per-user policy that lets you control the behavior of the Offline Files and Folders feature in Win2K. Lets you set policy to disable use of offline folders altogether, force synchronization of offline files at logoff, and set event logging on offline files and folders. Policies in the User and Computer Configuration sections in the GPO are duplicated, and in most cases, the Computer Configuration policy overrides the User Configuration policy. For more information about this, see the Explain text for a policy.
Computer & User Configuration Network Network and Dial-up Connections	Per-machine and per-user policy related to the Internet Connection Sharing (ICS) feature and to the use of Remote Access Service (RAS). The per-machine policy lets you disable ICS for computers subject to this policy. The per-user policy lets you control what users can do with LAN and RAS connections.
Computer Configuration Printers	Per-machine policy that lets you control how printer objects are published, browsed, and located in AD. This policy is used in conjunction with AD sites and location information on subnets to let your users locate printers based on the site they belong to.
User Configuration Start Menu & Taskbar	Per-user policy that lets you control all aspects of the Start menu. Included in this policy section are, among others, settings for removing the Run, Search, and Help dialog boxes from the Start menu as well as forcing personalized menus on or off.
User Configuration Desktop (includes Active Desktop and AD policies as well)	Per-user policy that controls the behavior of the user's Desktop and Active Desktop features, if enabled. The Active Directory folder in this policy lets you set the behavior of the AD Search dialog box that can be opened from the Start menu. Examples of user desktop policy include removing all icons from the desktop and removing the My Network Places and Internet Explorer icons from the desktop.
User Configuration Control Panel	Per-user policy that lets you hide the Control Panel, or specific applets in the Control Panel, from the user.
User Configuration Control Panel Add/Remove Programs	Per-user policy that lets you control the behavior of the Add/Remove Programs interface, including the ability to prevent the user from searching outside your organization for new applications and updates. You can also hide the Add/Remove Programs option altogether, effectively disabling the applet.



This Policy	Provides Policy For This
User Configuration Control Panel Display	Per-user policy that lets you control the options available in the Display applet (a Control Panel), including screen saver and background bitmap settings. You can also hide the tabs on the Display applet (for example, Settings, Background, and so on) to prevent users from modifying those settings.
User Configuration Control Panel Printers	Per-user policy that let you disallow users from adding printers. This policy also includes a neat feature that lets you specify a Web site where users can find lists of available printers on your network. If you enable this feature, it's an option added to the Add Printer wizard (choose Start>Settings>Printers).
User Configuration Control Panel Regional Options	Per-user policy that lets you restrict which languages a user can use.

Table 3.1: Administrative Template policies and their purposes.

As you explore the various Administrative Template policies that ship out of the box, you'll notice that quite a few policies overlap each other. The way to work around this is to carefully read the text on the Explain tab accompanying each policy item. In a lot of cases, the Explain text will point out other similar policy and let you know which take priority.

Software Installation

The Software Installation feature in Group Policy is, as I mentioned in Chapter 1, a way of distributing software by leveraging the Windows Installer engine that comes with Win2K. In this section, I'll walk through the process for deploying an application using this feature.

To leverage Software Installation your applications need to be packaged using the Windows Installer packaging format (that is, .msi files). The exception to this rule is that you can publish applications for individual users to the Add/Remove Programs Control Panel applet using a setup description file called Zero Administration Packaging (ZAP).

A .zap file is a text file that can be referenced by Software Installation to install an application that doesn't use the .msi packaging format. However, applications published in this manner don't accrue some of the benefits of .msi packaged applications. For example, you can't install .zap files using the escalated privilege feature, and they don't provide the life-cycle management features that the Windows Installer engine provides. Listing 3.2 shows an example of the format of a .zap file.

```
[Application]
FriendlyName = "ABC App 1.0"
SetupCommand = "p:\abc_app\setup.exe"
DisplayVersion = 1.0
Publisher = ABC Software Inc.
URL=http://abcsoftware.com
```

```
[ext]
```

```
ABC=
```

Listing 3.2: Viewing the format of a Zero Administration Packaging (ZAP) file.

In this listing, I've simply created a text file that lets me call the old-style setup application called setup.exe for an application called ABC App. In this file, the FriendlyName and SetupCommand key-value pairs are required fields, but the others aren't. The additional information will simply appear in the application information field in the Add/Remove Programs Control Panel applet when the application is published. The [ext] tag lets you associate a particular file extension with this published application.

In this example, if the user double-clicked a file with a .abc extension and the ABC App was not installed, it'd be automatically "faulted -in" and installed on the computer. The Software Installation feature lets you gain some of the benefits of "managed" applications without having to immediately repackage your application using the .msi format. Now that I've introduced the exception—.zap files—I'll talk about using .msi packaged applications to leverage the Software Installation feature in Group Policy.

 For more information on the Windows Installer technology, check out Chapter 3 of *The Definitive Guide to Windows 2000 Administration* at <http://www.realtimepublishers.com>. The site also contains links to other books that go into more detail about this technology.

Suppose I have a .msi package that I want to assign to users subject to a GPO that I've defined. My first step is to make that .msi package accessible to all users who might be installing it. In a Win2K AD environment, you can accomplish this in a number of ways. For example, you can put the package on a server share that is universally accessible to all of your users. This might be a good approach in a small network, but it could become a performance issue if you have hundreds of users installing the application across a worldwide network.

Alternatively, you can use Win2K's fault-tolerant Distributed File System (Dfs) to create a replicated share point across your AD infrastructure. In this case, the application setup package is placed under a fault-tolerant Dfs root and replicated across servers on your network that house replicas of that root. This approach is commonly used in conjunction with the Software Installation feature; the read-only nature of application setup packages lend themselves well to using Dfs.

 For more information on Dfs, read Chapter 5 of *The Definitive Guide to Windows 2000 Administration* by Sean Daily and Darren Mar-Elia (Realtimepublishers.com). You'll find a link to this book at <http://www.realtimepublishers.com>.

Lets assume that I've created my fault-tolerant Dfs root and have copied my .msi-based application setup packages to a Dfs folder. Now I'm going to walk through the process of assigning that application to a set of users using GPO-based Software Installation.

The first step is to open the Group Policy MMC snap-in, focused on the GPO you're interested in editing. Next, drill down into Computer Configuration|Software Settings|Software Installation. Right-click the Software Installation node, then choose the option on the context menu for

creating a new package. The familiar Open dialog box appears and lets you locate your .msi package.

Here it's important to enter a relative path to your .msi package rather than an absolute one. That's because the user processing the GPO needs to be able to find the package anywhere on your network. So don't use a drive letter designation (for example, d:\apps\mysetup.msi). Instead, use a relative path, ideally pointing to your Dfs share (such as [\\mycompany.com/dfsroot/appsetups/outlook2000/setup.msi](https://mycompany.com/dfsroot/appsetups/outlook2000/setup.msi)). Once you click Open, the Deploy Software dialog box opens, where you can choose Published, Assigned, or Advanced Published or Assigned (see Figure 3.21).

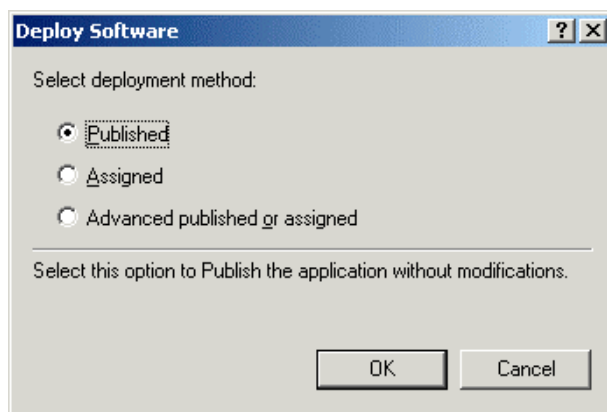


Figure 3.21: Viewing the options available when deploying a new application in Software Installation.

Unless I know that I want to make absolutely no customizations to my package deployment, I choose Advanced Published or Assigned. Then the Properties dialog box opens, shown in Figure 3.22.

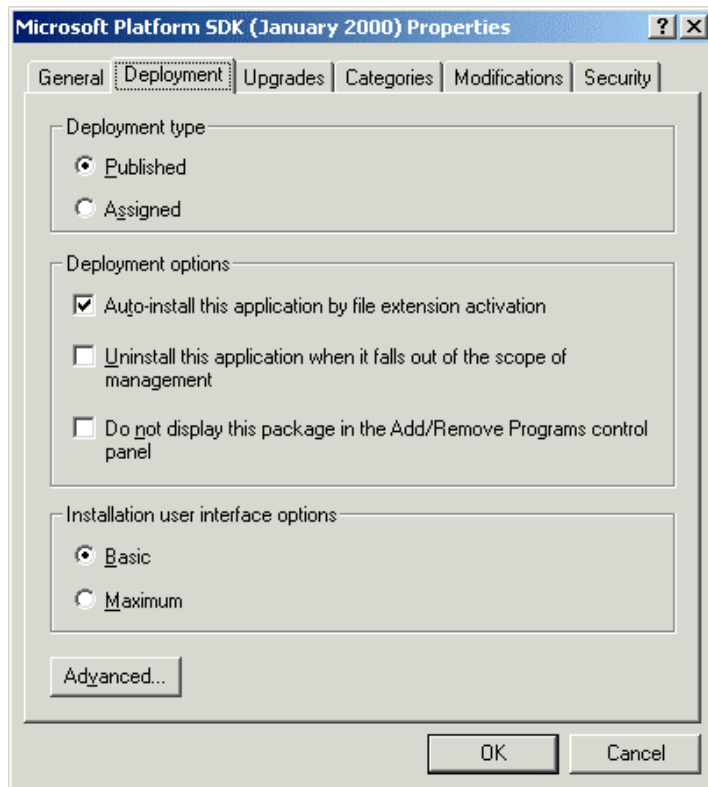


Figure 3.22: Viewing the Advanced Published or Assigned options in Software Installation.

The functionality provided by the six pages in this dialog box is described in Table 3.2

This Page	Does This
General	Provides basic overview information about the application name, version, and publisher.
Deployment	Lets you choose the type of deployment (Published or Assigned), deployment options, and UI verbosity. Under Deployment Options, if you choose to publish the application, you can also choose to have it install itself based on any file extensions that are associated with it. In addition, a check box indicates that if the GPO no longer applies to the user, you need to uninstall the application. You also have an option to prevent the application from appearing in the Add/Remove Programs Control Panel applet. This might be useful if you only want to assign the application using GPO, without giving the user the option to install it manually. You can also specify that, when the setup runs, it should show the user a basic or maximum UI. A basic UI shows minimal progress and requires no user interaction. If the package is coded to support it, the maximum UI presents the user with the full setup UI, including prompts that the user must respond to. Clicking Advanced on this tab displays the Advanced Deployment Options dialog box, shown in Figure 3.23. In this page, you can indicate that you want to ignore whatever language preferences may have been specified during setup. You can also specify that any instances of this application that were not installed by GPO should be removed. Note that this feature works only if the product code (shown in the Advanced Diagnostic Information section) for both the GPO-deployed application and the non-GPO-deployed application is identical. If it isn't, for whatever reason, Software Installation will simply install the application on top of

This Page	Does This
	existing installations. This page also includes some advanced diagnostic information on the package being deployed, including its product code (derived from the .msi package), a deployment count (which in my experience doesn't work), and the path in the GPT to the application assignment script (.aas file) that is generated when the package is deployed.
Upgrades	This page lets you set up upgrade relationships between the application that you're deploying and older versions of the application that were previously deployed using this GPO. Note that when you set up an upgrade relationship between, for example, ABC application v1.0 and ABC application v2.0, any users (or computers) that installed v1.0 will automatically receive v2.0. An upgrade package is different from a setup package in that it assumes that the 1.0 version has already been installed. Generally speaking, upgrade packages only update key files and registrations as needed. So if you set up an upgrade relationship between two versions of an application, it's important that the original version of the application remain available for users who haven't yet installed it. In that case, v1.0 will be installed, then v2.0 will be installed on top of it. If the new version of your software isn't a true upgrade to the previously deployed version, the best approach may be to remove the previously installed version (described below) and then republish or reassign the new version.
Categories	Categories let you define a way of grouping applications that appear in the Add/Remove Programs Control Panel applet. (They were described in Chapters 1 and 2.) Categories must be defined ahead of time before they can be selected from this page. To define categories, simply right-click the Software Installation node in a GPO in your domain. Choose Properties from the context menu, and you'll see the Software Installation Properties dialog box, shown in Figure 3.24. This dialog box presents some default options for Software Installation. If you click the Categories tab, you can define new categories, which will then become visible on the Categories page of your new package installation process.
Modifications	This tab lets you associate Windows Installer package transform files (.mst files) with a given setup. Transform files are custom modifications that you can perform on a default .msi setup package. Typical transform operations include choosing particular features to install in an application, changing the default installation folder for an application, or having an application install in Run from Network mode.
Security	Presents the familiar ACL editor interface. The Security tab gives you the ability to assign permissions on individual applications in a GPO. I mentioned earlier in this chapter that you can assign permissions on a whole GPO to filter the processing of that GPO to a particular group of users or computers. Well, you can actually filter the installation of particular applications by modifying the permissions shown in this dialog box. For a user or computer to "see" an application that has been deployed using GPO, that user or computer must be a member of a group that has Read permission on the application. By default, Authenticated Users has Read permission on any application that is deployed using Software Installation. However, you can remove this ACE (just as we did above in "Using Security Groups to Filter the Processing of GPOs") and grant Read permissions to subsets of user or machine groups. This gives you a very fine-grained control over who receives which applications, even in a single GPO. However, when it comes time to troubleshoot problems, this also increases the complexity of the GPO by an order of magnitude.

Table 3.2: Software Installation page options and their functions.

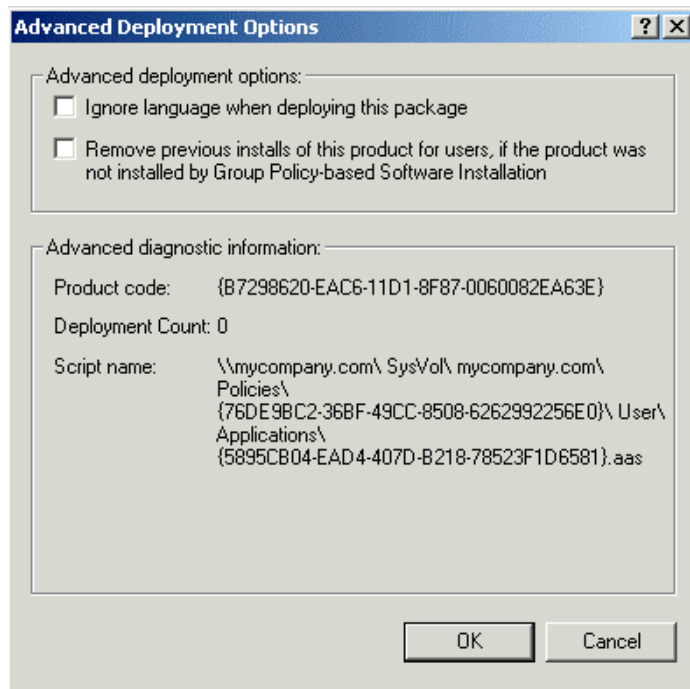


Figure 3.23: Viewing the advanced options available on the Deployment page in Software Installation.

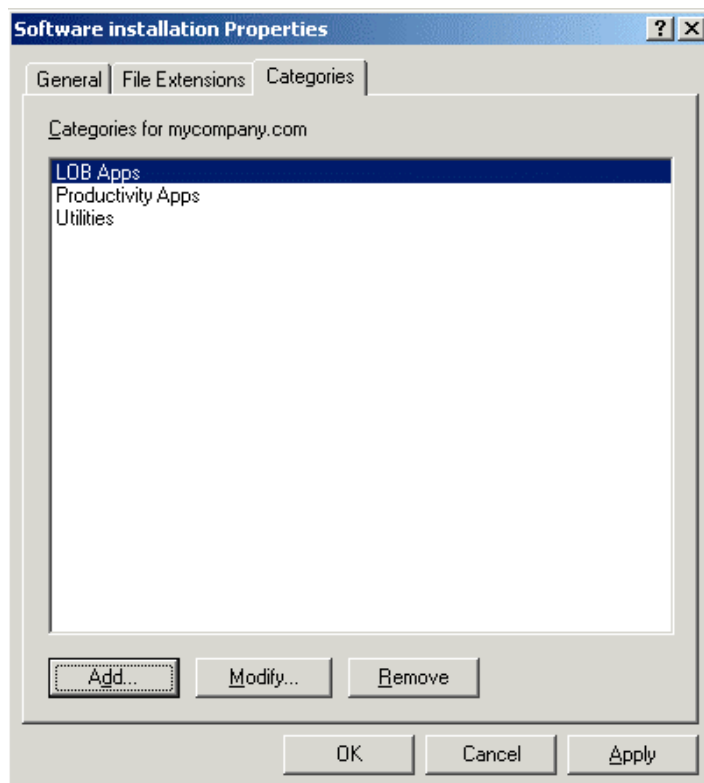


Figure 3.24: Viewing the Categories page of the Software Installation Properties dialog box, where you can create Software Installation categories.

I've talked about how you can install a new application using Software Installation, but how about if you're ready to retire an application? Well, it's just as easy to remove a previously installed application. In the GPO snap-in, right-click the application that you want to remove. Choose All Tasks from the context menu, then choose Remove from the submenu. The Remove Software dialog box will appear, shown in Figure 3.25.

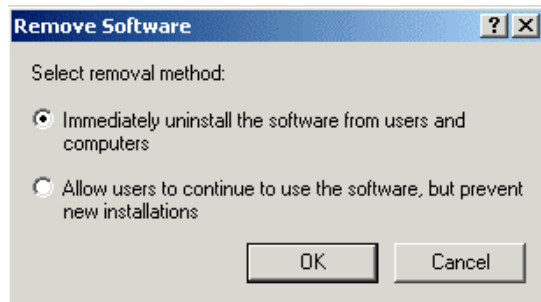


Figure 3.25: Viewing the options for removing a previously deployed application.

This figure shows two options. Either you can remove the application and have it uninstalled from all users and computers who received it, or you can simply leave it on the computers of those users who previously installed it and prevent any future installs. In either case, the application is no longer listed in the GPO and is no longer available for installation. There is also a Re-deploy Application command on the All Tasks submenu. This command tells the GPO to trigger a new installation on any computers or users who are subject to this GPO as though it were the first time the application was being deployed.

The Re-deploy Application option is useful if you've changed one or two files in an application setup and don't want to have to repackage it or set up an upgrade relationship between the old and the new versions. By re-deploying the application, the "patched" application is re-installed, and your new files are distributed to all users of the application. This approach is different from the one used by Windows Installer patch files, which are actually separate file packages with a specific purpose and relationship to the original .msi file. **Folder Redirection**

The Folder Redirection feature in Group Policy is an enhanced version of what was available in NT 4.0 system policy. Keep in mind that Folder Redirection is a user-specific policy only—it has no relevance for computers. Also, if the workstation processing a GPO detects a slow network link, Folder Redirection isn't processed. Folder Redirection lets you control the following user-profile-based folders:

- Application Data
- Desktop
- My Documents (and My Pictures in My Documents)
- Start Menu

Folder Redirection supports two types of redirection—basic and advanced. In basic redirection, you specify a server share to which you want to redirect the selected user profile folder. Users who process the GPO are then redirected to that single server share. This approach works best

when you're redirecting parts of the user profile, such as the Start Menu, to read only data that you want all users to see.

Setting Up Advanced Folder Redirection

The advanced redirection option allows you to specify a different server share, one that is conditional on the user's group membership. Figure 3.26 shows an example of how you might set up advanced folder redirection.

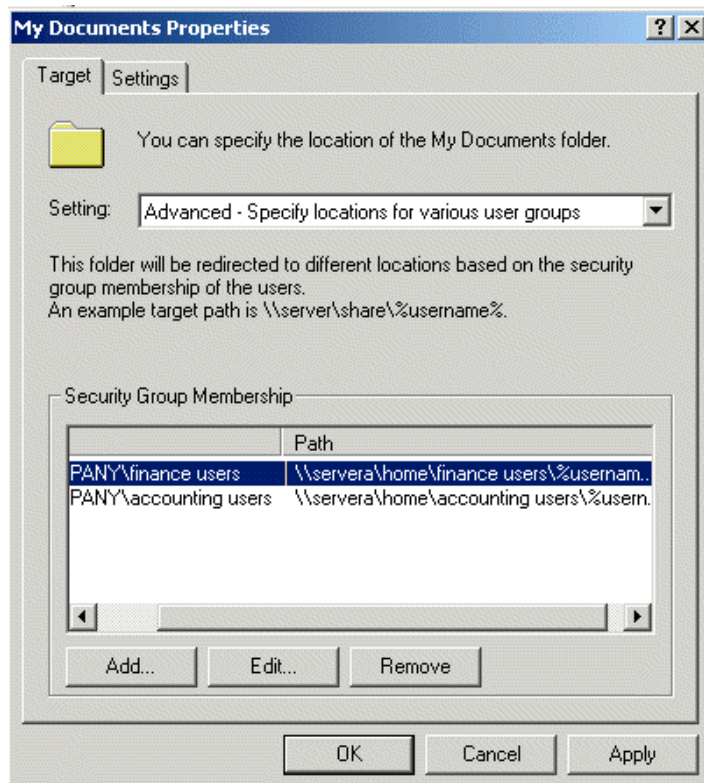


Figure 3.26: Using Advanced Folder Redirection in a GPO.

As the figure shows, I've redirected the My Documents folder to two different shares—`\\servera\home\finance users\%username%` and `\\servera\home\accounting users\%username%`—based on membership in the Finance Users and Accounting Users groups.

 Folder Redirection in Win2K doesn't support using mapped drives or environment variables (other than %username%) when you specify a path to a server share for redirection. You'll need to use UNC paths or local drive letters instead. I suspect this issue will be resolved in future releases of the product because it represents a step backwards in functionality from NT 4.0 system policy and the Custom Shared Folders feature.

The options available on the Settings page are shown in Figure 3.27.

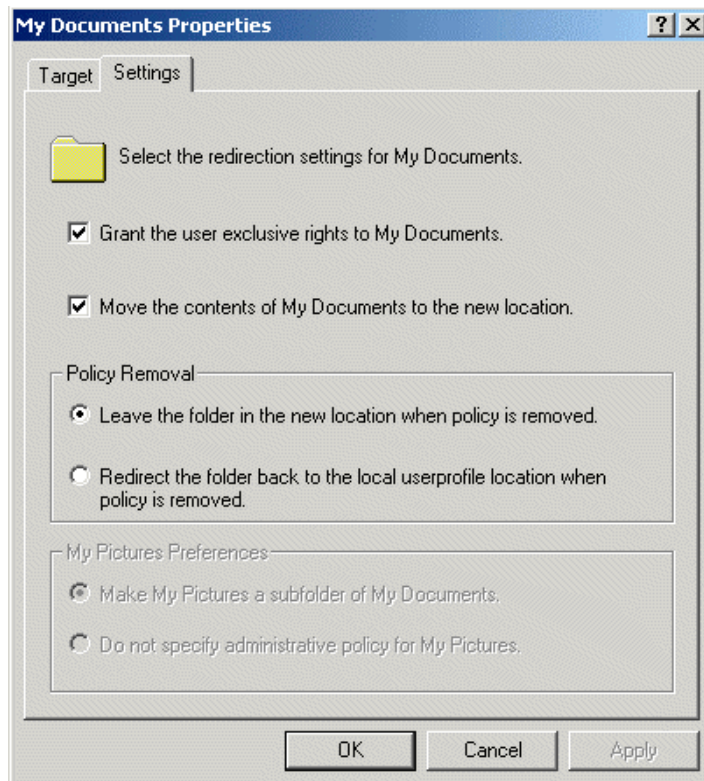


Figure 3.27: Viewing the Settings options for Folder Redirection.

The Settings page gives you the ability to grant exclusive rights to the folder that is being redirected. This option is only relevant for My Documents, Application Data, and Desktop redirection (not Start Menu). However, if you use it, be cautious. It's really only meant for folders that are user-specific. For example, if you're redirecting everyone's Desktop folder to a single server share, you wouldn't want to select this option. This would assign the server share granting the first user who processed the redirection exclusive rights to the folder. This option actually changes NTFS file system permissions to enforce the exclusion, so be careful!

The next option tells Folder Redirection that when redirecting a folder for the first time, it should move any files and subfolders in that folder to the server share that has been specified. This is generally desirable to do unless you're sure you don't want to bring old content along with your users. However, if you don't select this option and your users have critical files stored in the user profile version of their folders, they may not be able to easily retrieve them after redirection occurs: The folder will be pointing at a server share. You can specify that a folder be left in the new location or be redirected to the user profile location.. This is an important distinction. Deleting the GPO or denying the user access to it isn't the same as removing the folder redirection policy. To remove Folder Redirection, you actually have to undo folder redirection policy on this GPO. The next time the user processes the GPO, it'll behave as you specified. Namely, it'll either leave the folder where it was redirected (for example, a server share) or copy the contents of the redirected folder back underneath the user profile folder where it belongs.

Take my word for it—removing folder redirection can be complicated. You need to thoroughly test the process you want to use to undo folder redirection before implementing it, or the consequences may be very undesirable.

Security Settings

In this section, I want to make you comfortable with setting security policy in GPOs and knowing what to expect when you do. To start with, as was indicated in Chapter 2, there is a relationship between the security settings nodes in a GPO and the security templates on the one hand and the Security Configuration and Analysis MMC snap-ins that come with Win2K on the other. GPO security settings simply apply the security templates that you create using these other tools to many computers at a time. GPOs provide the transport, if you will, for enforcing organization-wide security policy. You can do it in real time by opening a GPO for editing and setting the options in it manually. Or you can pre-create a security template that contains all the settings you need, then import that template into a GPO. It's the latter process that I want to describe here.

About Security Templates

To begin with, all Win2K devices come with a standard set of security templates, usually located in %systemroot%\security\templates. Figure 3.28 shows an example of these template files.

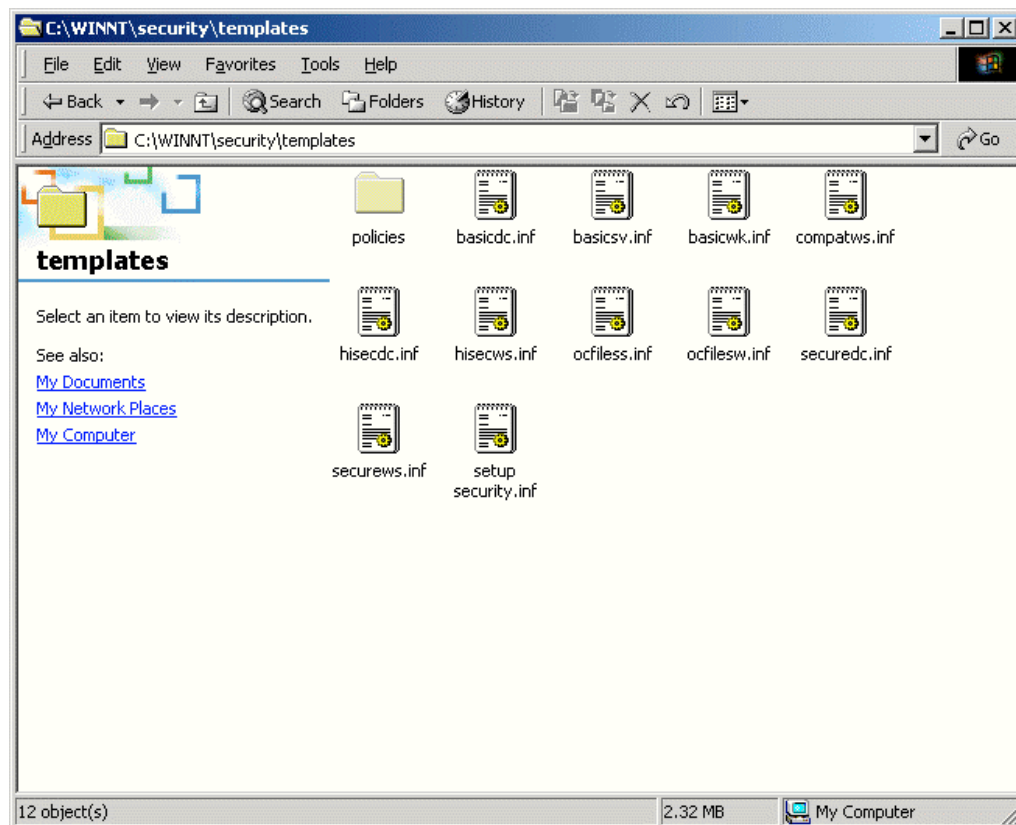


Figure 3.28: Viewing the security templates that come with Win2K.

These templates are simple text files that use a particular format for specifying each of the security settings that you see supported in computer-based GPO security (for example, audit policy, user rights, file system security, and so on). The Security Templates MMC snap-in that comes with Win2K lets you edit one of these existing template files or create one. In either case,

if you know the syntax, you can edit these files using a tool like Notepad. However, using the Security Templates MMC tool is easier and safer.

Let's suppose I want to create my own security template. It will set a number of audit policy settings that I want to import into a GPO. To begin with, I'll load the Security Templates MMC snap-in. When I open the tool and expand the top-level node, it automatically loads the templates located on the computer where I'm running the tool into the %systemroot%\security\templates folder.

Creating Security Templates

To create a security template, I simply right click the path listed and select New Template. I'm prompted to supply a name for my template and some descriptive text. I'll call this template Darren's Audit Policy. Once I've created the template, if I expand the folders underneath it, I see settings very similar to what is available in the computer-specific security settings in a GPO. Figure 3.29 shows an example of this.

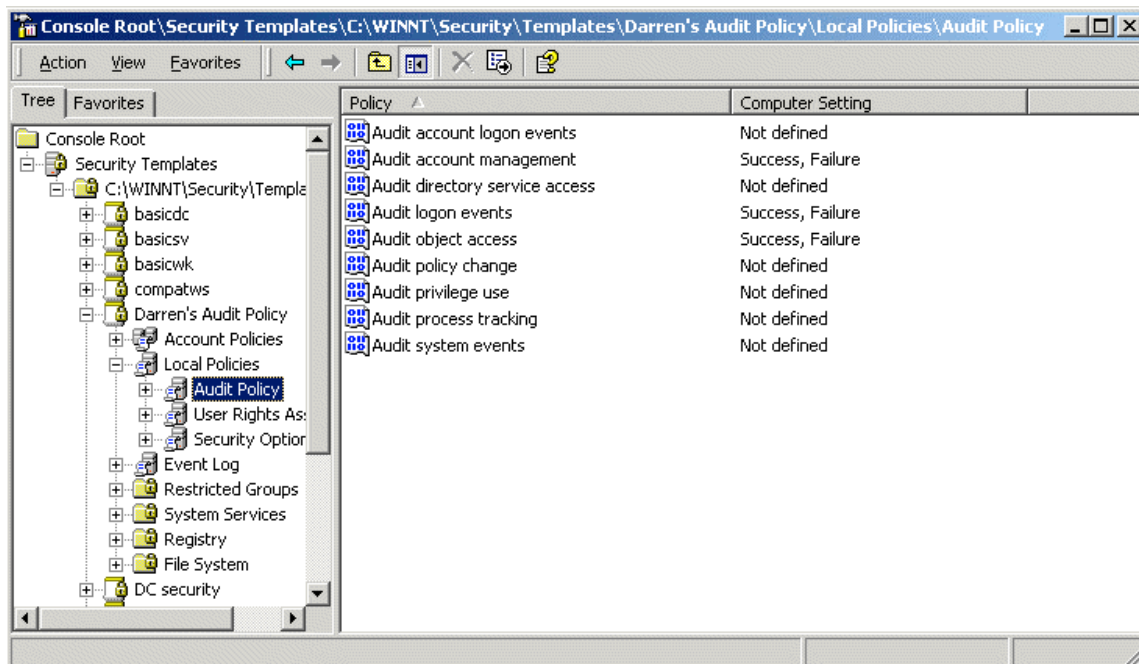


Figure 3.29: Creating a custom security template.

As you can see, I've drilled down to the Local Policies|Audit Policy section of the template and selected a number of audit policy settings. Once my changes are made, I save the changes to my template file by right-clicking the template name and choosing Save from the context menu. (Template files have a .inf extension.)

Importing Security Templates into a GPO

Now all I need to do is import my new template into the GPO where I want to enforce my audit settings. I open the Group Policy snap-in, focused on the GPO I want to edit, then drill down to Computer Configuration|Windows Settings|Security Settings. I right-click the Security Settings

node, then choose to Import Policy from the context menu. In the Open dialog box, I select my new template. Once it loads, if I drill down into the audit policy on that GPO, I'll see my changes made exactly as specified in the original template file. Figure 3.30 reflects this.

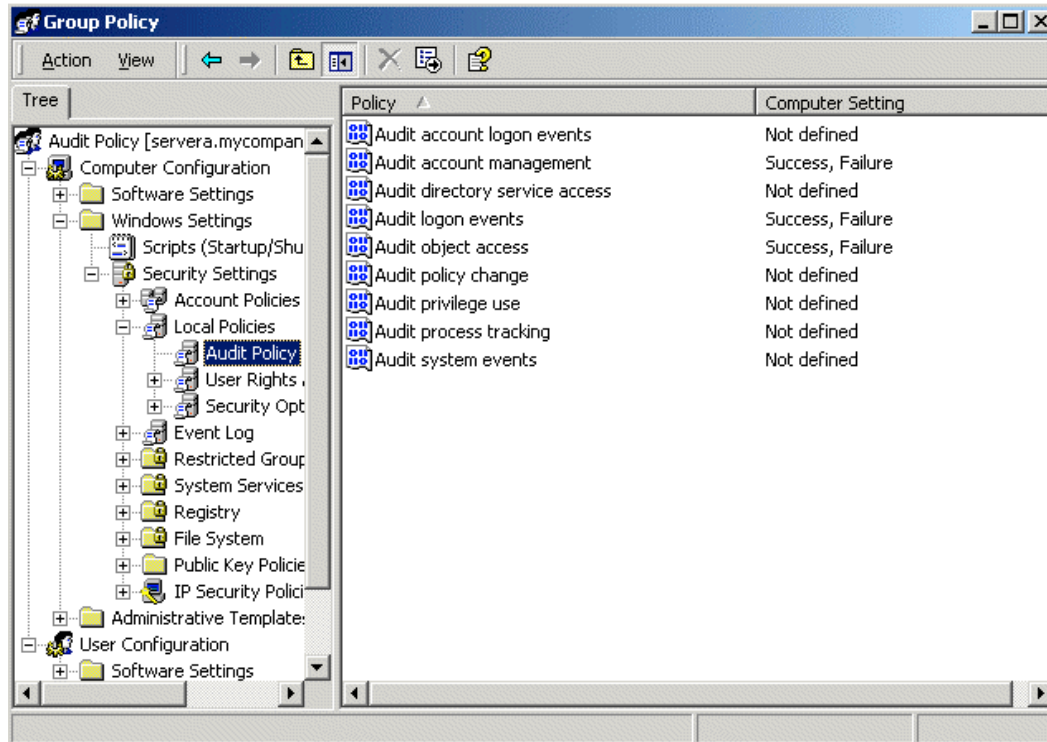


Figure 3.30: Viewing a GPO after importing a security template into the security policy settings.

As soon as I import the new policy template file into the GPO, the security settings become active, and the next computer to process the GPO will receive those new settings. In addition, if I make changes to other GPO security settings not specified in the template file, then reload the original template, the template file will overwrite any changes I made in the GPO itself. However, using templates as the definitive source for all GPO security settings, you build in a bit of change control that editing a GPO itself doesn't necessarily provide. You can think of security templates as the .adm files of GPO security.

Summary

In this chapter, I discussed creating and editing GPOs. I started by talking about how GPOs are linked to containers and discussed simple editing of policy. Then I moved into a detailed review of the options available to you when you edit a GPO. From there, I described topics such as filtering GPO processing using security groups. I then discussed in detail how some of the more popular features in Group Policy—Administrative Template Policy, Software Installation, Folder Redirection, and Security Policy—can be used to create useful GPOs.