INCLUDES FULL COVERAGE OF WINDOWS VISTA

Group Policy: Management, Troubleshooting, and Security

For Windows Vista™, Windows® 2003, Windows® XP, and Windows® 2000

Jeremy Moskowitz

Mark Minasi Windows® Administrator Library

SYBEX® SERIOUS SKILLS.

Contents at a Glance

5

ADM and ADMX Template Management

You might occasionally wonder, "Where do all those bazillions of policy settings come from?" They're not a magical gift from the Group Policy Fairy Godmother—they're encoded inside files within Windows which allow you to do the stuff you want to do. Group Policy has lots of nooks and crannies in which many options can be set. It's likely you'll spend most of your time manipulating the Administrative Templates section. Consequently, you need to know where all these settings come from.

In Windows XP, the Administrative Templates section comes from ADM files. In Windows Vista, that same section comes from ADMX files.

These templates hold the key to a large chunk of what makes Group Policy great. These settings are so important and powerful because they alter the Registry on the target computer. These ADM and ADMX files describe the Registry settings that can be toggled on or off through the Group Policy Object Editor.

That's where the duality of this chapter comes in. It's not only going to be really important to understand the "under the hood" goings-on of both ADM and ADMX files, but also where Windows Vista brings new features to the table when ADMX files are used.

And, while not strictly necessary, if you want to take your Group Policy game to the next level, you might want to invest some time in understanding the language used to create ADMX files. After you understand the syntax, you can create, modify, and troubleshoot almost any Registry change that is implemented by the Administrative Templates in the Group Policy Object Editor.

Finally, a little later in the chapter, we'll explore a new tool from Microsoft and FullArmor which can take existing ADM files you might already have and convert them into ADMX files.

You'll find the complete reference for creating your own ADMX templates in the "ADMX Template Syntax" download on this book's website. If you still feel you need to create ADM files after reading this chapter, the previous edition's download of a similar reference, "ADM Template Syntax," should still be available on GPanswers.com as well.

Policies vs. Preferences

One of the most heralded benefits of moving away from your old Windows NT 4–based System Policy is the nonpersistence of the Registry changes using Group Policy. Every Windows NT 4 System Policy change was *persistent*. When you enabled a System Policy, it stayed turned on until you set an explicit policy to turn it off. You couldn't just delete the policy and have the setting go away, as is the case with today's Group Policy engine. If you used Windows NT System Policy, you had to fight the same problem over and over.

Versions of Windows since Windows 2000 utilize a new model for policies. Microsoft created special locations in the Registry for Windows 2000, aptly named *Policies*. Microsoft documentation states that four Registry areas are considered the approved places to create policies out of Registry hacks:

- `HKLM\Software\Policies` (computer settings, the preferred location)
- `HKLM\Software\Microsoft\Windows\CurrentVersion\Policies` (computer settings, an alternative location)
- `HKCU\Software\Policies` (user settings, the preferred location)
- `HKCU\Software\Microsoft\Windows\CurrentVersion\Policies` (user settings, an alternative location)

These locations are preferred because they have security permissions that do not allow a regular user to modify these keys. Again, the preferred locations are noted above, if any software developers are reading this book (and you know who you are).

When a policy setting is set to "Enabled" and the client embraces the Group Policy directives, a Registry entry is set in one of these keys. When the GPO that applied the keys is removed, the Registry values associated with it are also removed. However, it should be noted that the application (or operating system component) needs to look for changes to these keys in order for it to take effect. That is, the Group Policy engine doesn't "notify" the application—the application has to do its own checking. So, with this in mind, if an older operating system receives a policy setting for a newer operating system, nothing "bad" happens. It just gets ignored.

It should be noted that local administrators have security permissions to these keys and could maliciously modify delivered GPO settings because of rights within this portion of the Registry.

This is the magic that makes Group Policy shine over old-style NT 4 System Policy; that is, Group Policy won't tattoo because it's being directed to go in a nonsticking place in the Registry. Old-style NT 4 System Policy had no such facility. Today, Microsoft calls these NT-style policies that tattoo *preferences*.

You might want to control a pet application that you have deployed in-house, say, DogFoodMaker 6.1. Great—you've decided you want more control. Now, you need to determine which Registry values and data DogFoodMaker 6.1 understands. That could take some time; you might be able to ask the manufacturer for the valid Registry values, or you might have some

manual labor in front of you to determine what can be controlled via the Registry. You'll then be able to begin to create your own templates.

However, after you've determined how DogFoodMaker 6.1 can be controlled via the Registry, you'll find you have two categories of Registry tweaks:

- Values that fit neatly into the new Policies keys listed earlier
- Values that are anywhere else

You'll have some good news and some bad news. If DogFoodMaker 6.1 can accept control via the Registry, you can still create template files and control the application. The bad news is that if the Registry punches it accepts are not inside the Policies keys listed earlier, you will not have proper Policies. Rather, they become old-style tattooing preferences.

To reiterate, the target applications must be programmed to look for values in the Policies keys. Some applications, such as Word 2000, check the Policies keys (specifically `HKEY_CURRENT_USER\Software\Policies\Microsoft\Office\9.0\Word\`).

Other applications, such as WordPad, do not "understand" the Policies keys. (WordPad looks in `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Applets\Wordpad`). Hence, WordPad wouldn't be a candidate to hand-create a template file for the purpose of coding for true policies settings. You could, however, still create your own *preferences* for WordPad that modify and tattoo the Registry. Therefore, you will have to do the legwork to figure out if your applications are compatible with the new Profiles keys.

Because preferences and policies act so differently, you will need to quickly identify them within the Group Policy Object Editor interface. You will want to note whether you're pushing an actual new-style policy to them or a persistent old-style policy. You'll see both cases in this chapter.

When viewed on Windows Vista, new-style policies are designated by little "paper" icons. When viewed on Windows XP, new-style policies are designated by little blue dots. Again, these are "proper" because they modify the Policies Registry keys.

Policies that represent Registry punches in places *other* than the preferred Microsoft policies are designated another way. On Windows Vista, they're represented by paper icons with a down arrow. On Windows XP, they're designated by red dots.

Again, you'll see this distinction a bit later as you work through the examples.

Since this is an important distinction in the rest of this chapter, let's recap:

- New-style policies are temporary Registry changes that are downloaded at logon and startup (and periodically in the background). They don't tattoo the Registry (though they are maintained and stay persistent should the user log on while offline). These are set to modify the Registry in specific Microsoft-blessed Policies keys. Applications need to be coded to recognize the presence of the keys in order to take advantage of the magic of policies. In the Vista Group Policy interface, these look like little paper icons.
- New-style policies also don't overwrite user preferences if they exist. For instance, if a program like Microsoft Word is policy-enabled, and a user specifies to enable "Correct two initial capitals," but later the administrator chooses to disable this setting with a GPO, the original user's desires will just magically "come back." So, when you remove a true policy setting from a GPO (that is, set to "Not Configured"), the original user preference will be "returned."

- Old-style preferences are persistent Registry changes sent from on high using the Group Policy Object Editor. These typically tattoo the Registry until they're specifically removed. In the Vista Group Policy Object Editor interface, they look like paper icons with a little down-arrow. Unlike new-style policies, if you remove the GPO, you "orphan" the settings on the target computer (no fun at all). They work like old-style NT System Policy. These can be set to modify the Registry anywhere.

Hang tight, dear reader. The differences between preferences and policies will be underscored a bit later when we add in additional templates and you create your own settings to manipulate your clients later in this chapter.

ADM vs. ADMX File Distinction

Because Windows XP and Windows Vista have such radically different ways of presenting (what appears to be) the same stuff to you via the Group Policy Object Editor, it might be helpful to get a brief rundown of each technology. It's likely you have a mixed environment—of Windows XP and Windows Vista. So, as we proceed in the chapter, you'll have a feel for what's going on under the hood.

Again, many settings are available in both the Computer and the User Administrative Template sections of the Group Policy Object Editor. How these settings are displayed depends on what is inside the default ADM templates. Therefore, when you create any new GPO, you start with baseline policy settings.

Windows XP ADM File Introduction

The default templates are stored in the `%systemroot%\inf` folder, which is usually `C:\windows\inf`, and you'll find the following templates are installed by default on Windows XP+SP2 machines:

- `Conf.adm`
- `Inetres.adm`
- `System.adm`
- `Wmplayer.adm`
- `Wuau.adm`

These five ADM templates create both the Computer and User portion within Administrative Templates of a default Group Policy. Table 5.1 provides information about what each default template is and what lives inside it.

`inetcorp.adm` and `inetset.adm` are two ADM templates which can alternatively be used to manipulate Internet Explorer settings. However, it is not advised, as they don't work well for newer versions of Internet Explorer.

TABLE 5.1 Default ADM Templates

ADM Template	Features	Where to Find in Interface
`Conf.adm`	NetMeeting settings.	Computer Configuration/User Configuration ➢ Administrative Templates ➢ Windows Components ➢ NetMeeting
`Inetres.adm`	Internet Explorer settings, including security, advanced options, and toolbar settings. It is equivalent to the options that are available when using the Internet Options menu inside Internet Explorer.	Computer Configuration/User Configuration ➢ Administrative Templates ➢ Windows Components ➢ Internet Explorer
`Inetcorp.adm` (not used in a GPO by default)	Used for Internet Explorer Maintenance preference mode settings.	User Configuration ➢ Windows Settings ➢ Internet Explorer Maintenance. We won't be exploring this particular ADM template much. If you want more information on its ins and outs, read `http://tinyurl.com/z3cae`. It is not suggested to use this unless especially directed by a specific Microsoft document or PSS person.
`Inetset.adm` (not used in a GPO by default)	Internet Explorer "Advanced Settings" for Internet Explorer 6.	User Configuration ➢ Windows Settings ➢ Internet Explorer Maintenance ➢ Advanced. Only visible in Internet Explorer Maintenance Preference mode (see Chapter 6's "Internet Explorer Maintenance Policies" section). It is not suggested to use this unless especially directed by a specific Microsoft document or PSS person.
`System.adm`	Operating system changes and settings. Most of the Computer and User Administrative Template settings are in this ADM template.	Everything else under Computer Configuration/User Configuration ➢ Administrative Templates
`Wmplayer.adm`	Windows Media Player 9 settings.	User Configuration ➢ Administrative Templates ➢ Windows Components ➢ Windows Media Player
`Wuau.adm`	Controls client's access to Software Update Services servers.	Computer Configuration ➢ Administrative Templates ➢ Windows Components ➢ Windows Update

Windows Vista ADMX File Introduction

As we saw with Windows XP, there's a mere handful of ADM files which make up the bulk of our settings. In Windows Vista, things change from ADM files to ADMX files, and what was once a handful is now an entire growler-full.

What's a growler? `http://tinyurl.com/jos6c`.

Windows Vista ADMX files are stored in the `%systemroot%\PolicyDefinitions` folder, which is usually `C:\windows\PolicyDefinitions`.

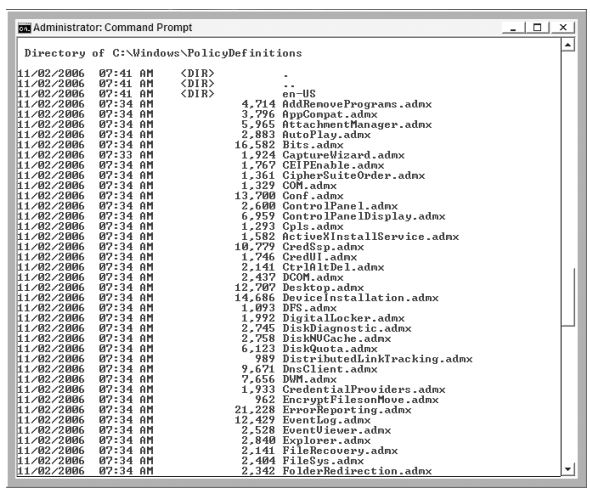
There are now about 132 ADMX files which roughly cover the same settings found in Windows XP. They're generally component specific. For instance, you'll find things like `WindowsMediaPlayer.admx` and `EventLog.admx`, amongst others.

Here's something neat about ADMX files—they're language neutral. That is, the definitions for the Registry values that are controlled are inside the ADMX file. However, the text strings describing the policy and the Explaintext are contained within a *separate* file called an ADML file. These ADML files are located in specific sub-directories for each language within the `c:\windows\PolicyDefinitions` folder. For instance, U.S. English is contained within the en-US directory, which can be seen in Figure 5.1.

en-US is for US English. For other locales, visit `http://tinyurl.com/qpomo`. For instance, HE is for Hebrew, RU is for Russian, DE is for German, AR is for Arabic.

On GPanswers.com in the "Book Resources" section, I'll have a table with the names of all the ADMX and ADML files, what they do, and more.

FIGURE 5.1 A quick list of some ADMX files. Note the language-specific directory here for English (en-US).



ADM vs. ADMX files—At a Glance

Our goal for the rest of the chapter is to give you an in-depth look at both ADM and ADMX files and for you to understand the differences between them. However, before we get going, here's a quick little reference table so you can see where we're going and also utilize this table as an ongoing reference.

ADM Files	ADMX Files
Lots and lots of definitions are packed into several large-ish files. The biggest one is `SYSTEM.ADM`.	Definitions are split logically into much smaller ADMX files, generally by Windows feature area.
Each ADM file contains settings in one specific language.	ADMX files are language neutral. Language-specific information is contained within a corresponding ADML file. Language-specific files live in hard-coded directories. For example, U.S. English language files live in `%systemroot%\PolicyDefinitions\en-us`.
Live on each Windows XP machine in `%systemroot%\inf`.	Live on each Windows Vista machine in `%systemroot%\PolicyDefinitions`
Every time a GPO is "born" it costs about 3–5MB on each Domain Controller because the ADM files are placed inside the GPO.	GPOs created from ADMX files never have big space requirements. That's because the ADMX files are never pushed into the GPO themselves (regardless if the Central Store is used or not). We'll discuss the Central Store a bit later.
Use their own proprietary ADM syntax for describing Registry policy.	Use standard XML as the syntax for describing Registry policy.

Creating and Editing GPOs in a Mixed Environment

As I noted in Chapter 1, Windows XP and Windows 2003 have about 200 more policy settings available to them than their Windows 2000 pals do. And Vista has about 700 more policy settings than Windows XP.

The good news (as I've previously stated) is that Windows 2000 clients ignore policy settings meant for Windows XP or Windows 2003. And Windows XP clients ignore policy settings meant for Windows Vista. This makes sense: older clients don't have the "moving parts" required to do anything if these policy settings are set.

You're likely to have a mix of client and server systems. It's likely you'll have:

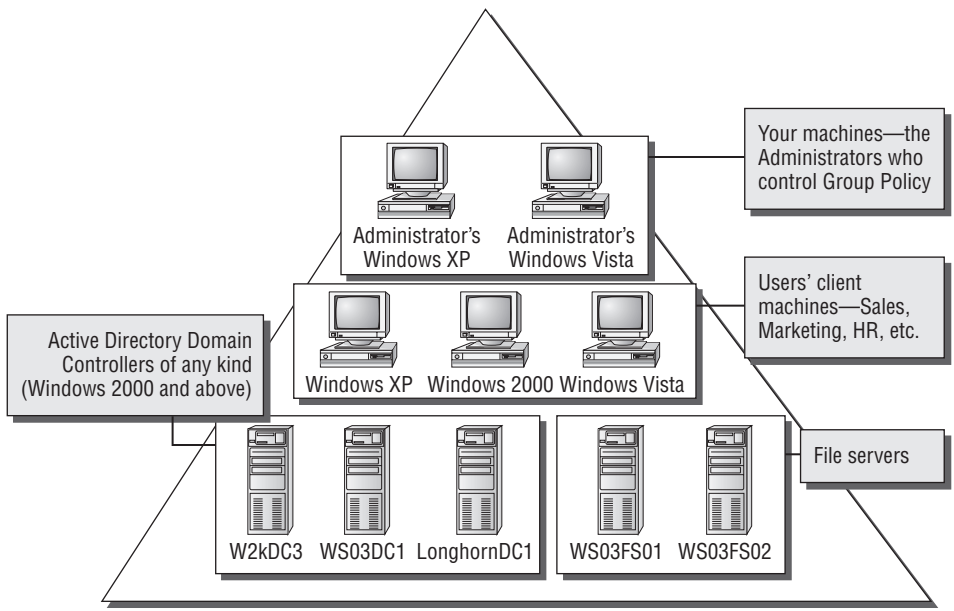
- Domain Controllers: Windows 2000, Windows Server 2003, and/or Longhorn Servers
- Member servers: Windows 2000, Windows Server 2003, and/or Longhorn Server
- Client machines (your users' machines): Windows 2000, Windows XP, Windows Vista
- Management stations (your machines, the ones you manage Group Policy from): Windows XP, Windows Vista

Figure 5.2 shows a typical Active Directory domain that could be representative of what you might have.

The question is: With all these types of client systems, how do we ensure we've got the maximum power to control them all?

That's what we're going to explore in this next section.

FIGURE 5.2 A typical Active Directory domain with administrative systems, client systems, Domain Controllers and servers



How Do You Currently Manage Your Group Policy Objects?

Before we proceed, you need to answer this question: How do you currently manage, create, and modify your GPOs?

- Do you walk up to a Domain Controller (or use Terminal Services to connect directly to a Domain Controller) to create or modify your GPOs?

- Do you use any machine you happen to be working on that day to create or modify your GPOs? (This could be a Windows 2000, Windows XP, server—whatever.)
- Do you use a specific machine to manipulate all the GPOs over which you have control? That is, do you have a *management station* you use when you need to manage your GPOs?

If you use either the first or second option, you're likely going to want to change your habits and start working with a strategy that gets you toward a *management station*.

Here's why. Every time a new operating system is released (and again each time a new service pack is released), there's more power to behold. Here's a brief history of increased power and what you can control:

Windows Version	Number of Policy Settings	What Can You Control?
Windows 2000	About 300 policy settings	Windows 2000
Windows 2000 + SP1	About 20 additional policy settings	Windows 2000, Windows 2000 + SP1
Windows 2000 + SP4	5 additional policy settings	All version of Windows 2000
Windows XP	About 150 additional policy settings	Windows XP and all versions of Windows 2000
Windows Server 2003	About 24 additional policy settings	Windows Server 2003, Windows XP, and all versions of Windows 2000
Windows XP + SP1	About 10 additional policy settings	Windows XP + SP1, Windows XP, and all versions of Windows 2000
Windows XP + SP2	About 600 additional policy settings	Windows XP + SP2, Windows XP + SP1, Windows XP, and all versions of Windows 2000
Windows Server 2003 + SP1	About 5 additional policy settings	Windows Server 2003 + SP1, Windows Server 2003, Windows XP + SP2, Windows XP + SP1, Windows XP, and all versions of Windows 2000
Windows Vista	About 700 additional policy settings	Windows Vista, Windows Server 2003 + SP1 , Windows Server 2003, Windows XP + SP2, Windows XP + SP1, Windows XP, and all versions of Windows 2000

Windows Version	Number of Policy Settings	What Can You Control?
Longhorn Server	Even more additional policy settings (not yet known at this time)	Longhorn Server, Windows Vista, Windows XP + SP2, Windows XP + SP1, Windows XP, Windows Vista, Windows Server 2003 + SP1, Windows Server 2003, and all versions of Windows 2000

Additionally, when service packs come out, Microsoft has been known to update the wording of policy settings and the Explaintext for clarity (though its underlying actions are usually the same).

So, Microsoft makes updates; you have more power, right? Sure. But the message is clear: if you want to control every client and server machine in your environment using Group Policy—use the latest version of the OS—Windows Vista.

What Happens When You Create a New GPO?

Windows XP uses ADM files and Windows Vista uses ADMX files (for editing GPOs)—and different things happen when GPOs are created using these management stations types. Figure 5.3 shows the "Reader's Digest" version of what we'll be discussing here, and after that will be the in-depth analysis of what's going on.

Creating and Editing a New GPO While Using a Pre-Vista Management Station

In order for us to create GPOs using a Windows XP machine, we need to have the GPMC console loaded (downloadable at `http://tinyurl.com/q77wx`). Note that the GPMC requires the .NET Framework files (downloadable at `http://tinyurl.com/758p8`).

Note the GPMC requires .NET Framework 1.1. If you *only* have 2.0, the GPMC won't install.

Recall from Chapter 4 that when you use any ADM templates, these templates are added to the file-based Group Policy Template (found in SYSVOL) of the GPO. Unfortunately, there's no master update location where you can just drop the latest ADM files from Microsoft (or other vendors) and universally update the ADM files of existing GPOs and any future GPO

that will be created. Indeed, you'll need to understand where new GPOs get their ADM templates from when you create new GPOs or modify existing GPOs.

In all cases, the editor you use (either Active Directory Users And Computers or GPMC) really uses the GPEDIT MMC snap-in (really the GPEDIT.DLL) when actually poking around or creating new GPOs. GPEDIT pulls the ADM template files from the computer it is running on. And it yanks these ADM template files from `%systemroot%\inf`—usually `c:\windows\inf`—directly into the GPO. Each time you do this, you're burning about 3–5MB of disk space—on every Domain Controller. This is because all material inside the GPO is replicated to every Domain Controller.

If you've created 100 GPOs using pre-Vista machines (like Windows XP or Windows 2000), you're using about 300–500MB of disk space—on every Domain Controller to store these ADM files. This problem is called SYSVOL bloat. In Figure 5.4, you can see a sample SYSVOL with several GPOs. Recall that GPOs live on every Domain Controller in the `sysvol\corp.com\Policies` directory underneath their GUID. Each GPO has an `ADM` directory each containing the same ADM templates at about 3–5MB each directory.

FIGURE 5.3 What's copied into the GPO when using which type of management station

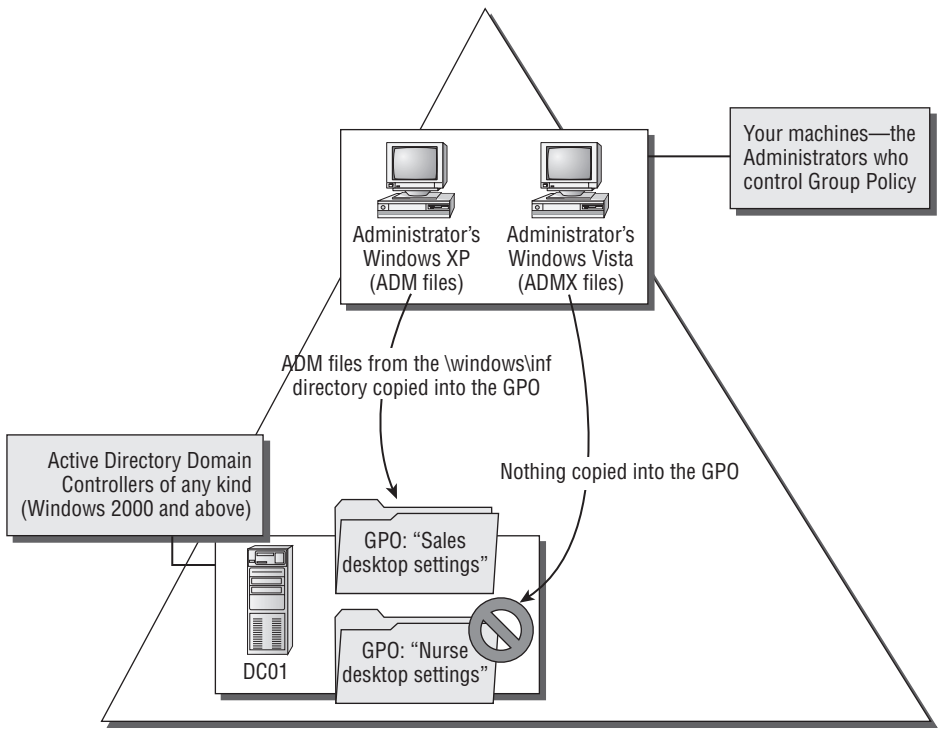
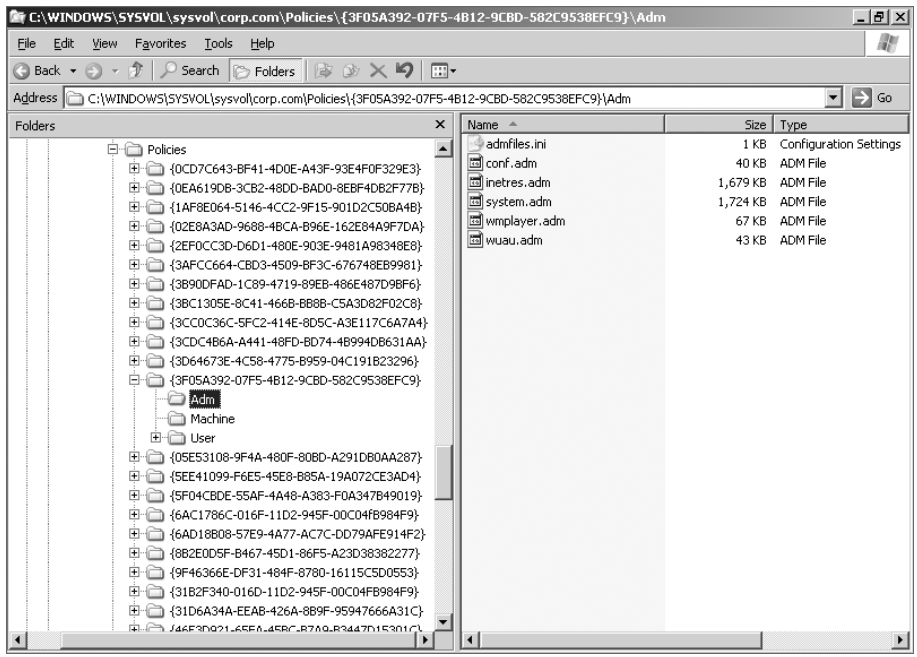


FIGURE 5.4 Every GPO created with a pre-Vista management station pushes about 3.3MB into SYSVOL.



How to Prevent SYSVOL Bloat If You're Still Using Pre-Vista Management Stations

There is a way to avoid copying up the ADM files into the GPO, hence wasting the 3–5MB on each Domain Controller per GPO. The trick is to use a policy setting entitled **Always use local ADM files for Group Policy Object Editor** (located in Computer Configuration ➢ Administrative Templates ➢ System ➢ Group Policy) and have it affect your management station.

By enabling this policy, you're telling your management station: "I'm not going to push ADM files into the SYSVOL folder." Sounds great, right?

The downside, however, is that if you try to edit the GPO on a machine that doesn't have the same ADM templates as the GPO (or worse, the local machine is just plain missing an ADM template), you simply won't be able to edit the GPO the way you want. You'll have to track down the original machine that had the full complement of ADM templates to properly manage the GPO.

Because of the downsides, this workaround is only suggested for very large environments that have lots of GPOs which are taking a long time to replicate because of all the ADM template data being pushed into the GPO.

Here's the big ol' scary warning about the policy setting: it only works if the management station is Windows Server 2003 (not Windows XP). Why? I have no idea. So, if you want to prevent SYSVOL bloat from ADM files, and you want to utilize this sneaky way to do it, you absolutely must make your management station Windows Server 2003 (and not Windows XP).

Microsoft talks a bit more about this in Knowledge Base article 816662.

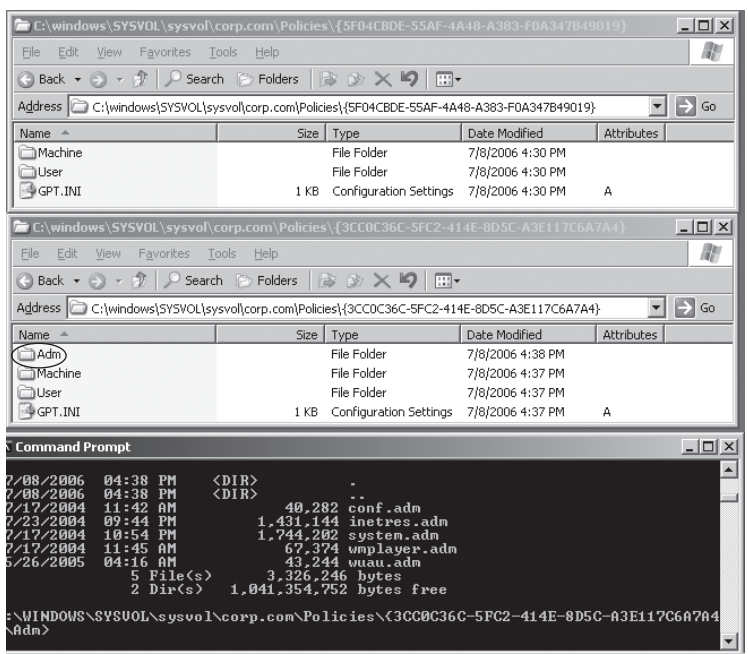
Creating a New GPO While Using a Windows Vista Management Station

In order for us to create GPOs using a Windows Vista management station, we don't need to do much. The GPMC is already loaded and waiting for us. We can just type **`gpmc.msc`** at the command prompt to fire it up.

Once we create a GPO using our Windows Vista management station, we can also take a look at what's generated inside SYSVOL. In Figure 5.5, you can see the top Window was created using a Windows Vista management station. You know this because there's no ADM directory.

And, because there's no ADM directory (and no ADM files inside it) there's no wasted space (SYSVOL bloat) from ADM files.

FIGURE 5.5 The top window shows a GPO's contents when created using a Windows Vista management station. The middle window shows a GPO's contents when created using a Windows XP management station. The bottom window shows the contents of the ADM directory for the GPO created using the Windows XP management station.



What Happens When You Edit an Existing GPO?

Here's where things get complicated. That is, you could have the four following situations:

- Scenario 1: Start out by creating and editing GPO on a pre-Vista management station (like Windows 2000, Windows XP, Windows Server 2003, and so on). Edit using another pre-Vista management station. In this scenario, no Windows Vista is involved.
- Scenario 2: Start out by creating and editing GPO on a pre-Vista management station. Edit using a Windows Vista station.
- Scenario 3: Start out by creating and editing GPO on a Windows Vista management station. Edit using another Windows Vista station.
- Scenario 4: Start out by creating and editing GPO on a Windows Vista management station. Edit using a Windows XP management station.

Scenario 1: Start Out by Creating and Editing GPO on Pre-Vista Management Station. Edit Using Another Pre-Vista Management Station.

Again, here, Windows Vista isn't involved. In this scenario, it's all about pre-Vista machines using old-school ADM templates and ADM template behavior. And, of course, note that by creating a GPO using a Windows XP machine, you won't be able to get to any of the Vista goodies—that's because all the Windows Vista goodies are only available when you use a Windows Vista management station.

So, let's imagine that you've created 128 GPOs using an old-and-crusty Windows 2000 machine. Of course, all 128 GPOs have the Windows 2000 versions of those ADM templates (yes, old and crusty).

Now, you learn about a policy setting in Windows XP that requires the corresponding Windows XP templates. What are you going to do?

Easy! Jump on a Windows XP machine and edit the GPO using the GPMC!

This is because, as we already understand, the ADM template files used to modify and update a GPO are always copied from your management station. Older ADM templates inside GPOs are automatically updated when you re-edit a GPO on a machine that has new ADM templates.

When you edit the GPO on your Windows XP management station and merely look at the policy settings in the Administrative Templates section, the editor will say: "Ah-ha! I've got Windows XP templates available to me! This specific GPO's ADM templates are only Windows 2000! I can tell because the date is sooo old. I'll update the underlying ADM templates automatically from `c:\windows\inf` in Windows XP—without even saying a word. That's because I have newer ones!"

And it then proceeds. And it proceeds because the time/date stamp for Windows XP ADM templates your editor has access to is more recent than the time/date stamp for Windows 2000 ADM templates. It's doing you a favor behind your back. You must repeat for every old GPO you want to update. If you want to update all your GPOs with Windows XP ADMs, you simply have to open each old GPO and look at the policy settings in the Administrative Templates section. But again, you need to do this from a Windows XP management station. Then they'll be updated.

Again, there's no universal master update location where you can just "drop in" your latest ADM templates and be done. However, with a script, you can update all your GPOs at one time (see the sidebar "Automatically Updating All Your Existing GPOs at Once with the Latest ADM Templates.")

Automatically Updating All Your Existing GPOs at Once with the Latest ADM Templates

In Chapter 7, you'll get a grip on all the myriad of things you can do with scripting and Group Policy. However, one thing that we won't tackle there (but we do want to tackle here) is how to automatically update all your existing GPOs with the latest ADM templates. As of this writing, the latest ADM templates are Windows 2003/SP1, but you could use this same tip to update all your GPOs with the ADM templates from, say, Windows XP/SP2 or earlier (not that you would really want to). Or, use this tip when XP/SP3 comes out.

To update all your GPOs (or just some of them), Microsoft has a downloadable script that will do this for you at `http://tinyurl.com/7v4s2`. It runs as a command line (as opposed to a GUI-based script). When you're ready to give the script a try, be sure to run it from the command line as `cscript admupdate.vbs` so it continues to use the command line for output (and not try to push data to the graphical output).

Here's what you need to tell the script:

- You need to tell it which GPOs to update. You can update using the /GUID switch, the /GPOfriendlyname switch, or the very powerful /ALL switch.
- You need to tell it where the latest ADM files reside. You do this with the /ADMSRC switch.
- You need to tell it what domain to update. You do this with the /DNSDOM:*<domain>* switch.

There are other switches available.

But, if you tell it just this much information, it performs a *simulation* of what it will do.

When you're actually ready for the script to do the deed and perform the upgrade, you need to add the /FILECOPY:ON switch (not shown in the preceding example). This actually performs the work. Note that this could take a *long* time and cause a *lot* of replication traffic. So, be sure to do it in the off-hours if possible.

Again, running this script isn't expressly necessary—for two reasons. First, because, as we've discussed, anytime you specifically touch an old GPO with an updated management station, the GPO will be automatically updated. Use this script to simply guarantee that the latest ADM files are pushed to every GPO. Second, by the time this chapter is over, I'm going to have convinced you to use a Windows Vista management station. And, then the GPOs themselves won't care about ADM files at all.

But, if you're still in a Windows XP-only environment, where you don't even have one Windows Vista management station, then this tip is still useful for you.

Scenario 2: Start Out by Creating and Editing a GPO on a Pre-Vista Management Station. Edit Using a Windows Vista Station.

This will be the common "upgrade" scenario. That is, you've already got your gaggle of GPOs created. You created them using Windows 2000, Windows XP, or Windows Server 2003 with Active Directory Users and Computers or the GPMC. Now, Vista comes along, and you're ready to use it. What happens?

Not much! If you start to use a Vista management station and edit an existing GPO created by a pre-Vista operating system, nothing happens in SYSVOL. No Vista ADMX files are copied anywhere, and very little happens overall.

However, while you're editing the GPO, you'll have access to all the latest-greatest Vista policy settings, one of which is seen in Figure 5.6.

For argument's sake, let's say you decided to enable "Turn off Windows HotStart"—a Windows Vista-only feature.

Now, what happens if you try to edit and/or report on those settings using Windows XP? Short answer: It's not good. That's because Windows XP doesn't know how to interpret the Vista-only settings you've set within the GPO. If you try to edit the GPO on a Windows XP machine, you simply won't see the newly available Windows Vista policy setting.

And, if you try to look at it using the GPMC's "Settings Report" feature, the Vista-only settings show up as "Extra Registry Settings" as seen in Figure 5.7.

In Figure 5.7 you can see the Settings tab from GPMC running on an XP machine, which is a report of what's going on inside the GPO.

FIGURE 5.6 Editing an existing GPO with Vista gives you the ability to see updated settings.

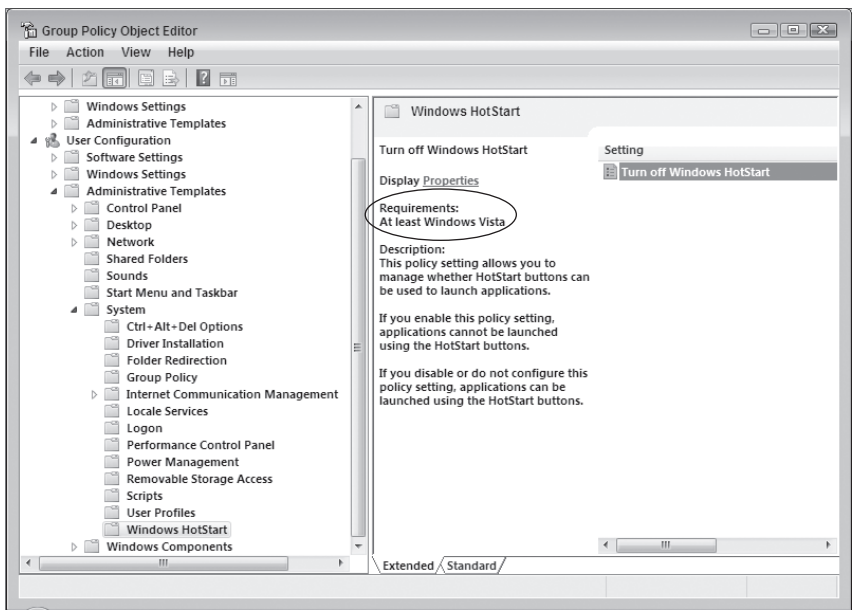
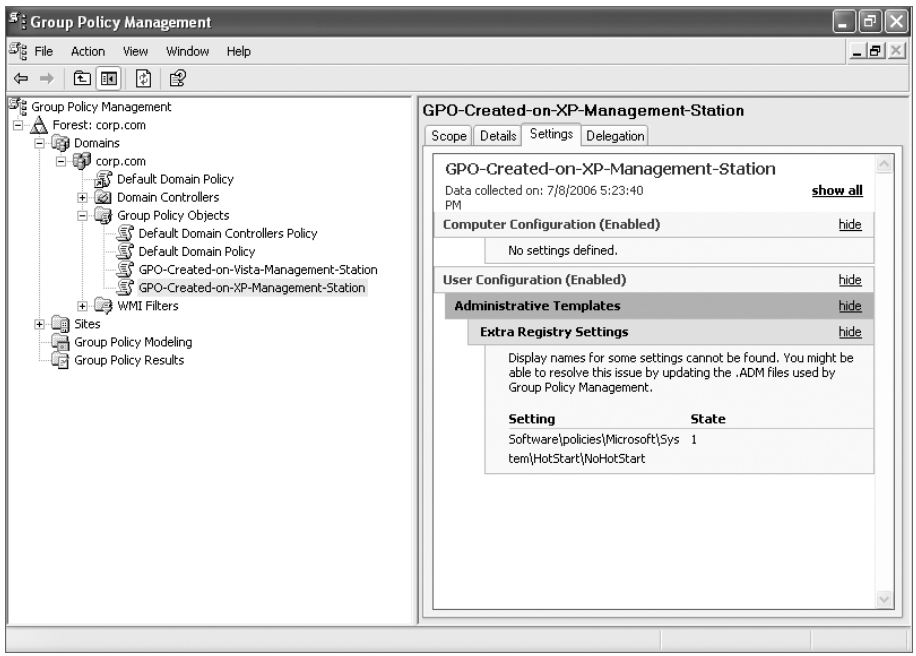


FIGURE 5.7 Windows XP doesn't know how to interpret Vista-only settings within a GPO. These settings show up as "Extra Registry Settings."



Again, if you were to continue to use your Windows XP management station to *edit* the GPO, you'd find that you simply wouldn't be able to find the "Turn off Windows Hotstart" policy setting—or any other Windows Vista-specific policy setting for that matter.

While it's clearly not a good idea, there is nothing that technically that prevents you from using Windows XP to make a change to a GPO that was edited by Vista. In short, you simply can't see the Vista settings.

If a custom ADM file has been added to the GPO (yes, ADM) then Vista will utilize it and present it.

Scenario 3: Start Out by Creating and Editing a GPO on a Windows Vista Management Station. Edit Using Another Windows Vista Station.

This is the scenario you want to strive for. That is, always use Windows Vista to create and edit your GPOs. When Windows Vista + SP1 comes out (with its additional policy settings),

you'll be all set. You won't have to do a thing (except use a Windows Vista + SP1 management station) to be "latest and greatest."

If you backtrack to a Windows Vista (no service pack) machine, you won't have access to the latest and greatest Vista + SP1 settings (and the GPMC won't accurately report what's Enabled or Disabled, either because it simply cannot "know" about those updated policies).

So, it sounds like the message is. "Always use the latest-greatest operating system and service pack as my management station." But what if you're not ready to personally upgrade your management station to the latest and greatest? Or, what if you had 20 administrators, each with their own management station?

If *only* there were a way to ensure that all your administrators always used the latest and greatest ADMX files, you'd have no issues. That way, even older version of Windows Vista would be able to determine the latest and greatest settings.

Sounds like a dream, right? Good news: It's a dream that we'll make a reality in the next big section, with the "Central Store." so stay tuned!

Scenario 4: Start Out by Creating and Editing a GPO on a Windows Vista Management Station. Edit Using a Windows XP Management Station.

Avoid this scenario whenever possible. This is the worst of all worlds because when you originally created the GPO on your Windows Vista management station, you did so without copying the 3–5MB of ADM files up (remember: Windows Vista doesn't use ADM files). So, you did good here!

However, by merely viewing the GPO using Windows XP, you end up pushing up the 3–5MB of ADM files into the GPO. So, every time you do this, you'll see an ADM directory inside the GPO because they were pushed up from your Windows XP machine. And it's done "invisibly."

Vista Management Stations and the Central Store

As we discussed, the ideal world is to use only Windows Vista as your management stations. Remember: if you have even one Windows Vista client machine out there in sales, marketing, Human Resources, and so on, you'll need to manage it *from* a Windows Vista machine, because Windows XP won't have the definitions of the policy settings that Windows Vista clients have.

So, we'll assume from here on that you'll only be using Windows Vista as your management station, eschewing Windows XP management stations. If you want the best practices for Windows XP management stations (again, only if you have zero Windows Vista clients, right?), then you'll have to pick up the previous edition of this book (same chapter, Chapter 5) which has lots of tips, tricks, gotchas, bugs, trials, and tribulations about Windows XP management stations.

As you're reading this right now, Vista is pretty darned new. But let's fast forward a bit and assume, oh, that we're up to Windows Vista + SP3. Yep, Windows Vista Service Pack 3 has just been released and you need to control the new whiz-bang features that only come with Windows Vista + SP3 client computers. (Again, I'm dreaming a little into the future here; new whiz-bang features might or might not come in service packs or other delivery vehicles, but stay with me through this example anyway).

"No problem!" you say, "I'll just create a Windows Vista management station." And you'd be right! Except that you already have a Windows Vista management station. So you wouldn't want to run out and create a whole new machine just for this. You'd want to leverage the Windows Vista management station you already have, right?

Sure!

This is easy! You're a diligent administrator (you bought this book, after all), and you know you have two ways to update your current management station:

- Apply SP3 to your Windows Vista management station. That would update the ADMX files which live in `c:\windows\PolicyDefinitions`.
- Or, you could forgo applying SP3 to your Windows Vista management station and simply copy the ADMX (and associated ADML files) from another Windows Vista + SP3 machine to your management station. Again, you'll plunk them in the `c:\windows\PolicyDefinitions directory`.

So, the message again sounds simple: whenever Microsoft has new ADMX/ADML files, get them into your management station.

Simple, yes—until you realize you have 20 administrators in your company, each with their own Windows Vista management station. Or you remember those administrators who love to bounce from machine to machine because they have three sites to run. Yikes! How are you going to guarantee that all of these administrators will use the updated ADMX files?

Let's assume you've successfully upgraded *your* Windows Vista management station to SP3, but only some of your 20 administrators successfully upgrade to Windows Vista + SP3 (or have created custom ADMX files) (or jam in the ADMX files into `c:\windows\PolicyDefinitions`).

This becomes a big problem—fast. Here's why: If you create a new GPO, that GPO will have the definitions for all the whiz-bang stuff Windows Vista + SP3 has to offer. However, when another administrator (who doesn't have the latest ADMX files) tries to edit or report on that GPO, they simply won't see the policy settings for Windows Vista + SP3 available.

GPMC reports about this newly created GPO would show them as "Extra Registry Settings," but actually trying to edit the GPO itself will not show the new whiz-bang features.

What you need is a way to ensure that all administrators who are using Windows Vista management stations have a one-stop-shop way to ensure they're getting the latest ADMX files. That way, everyone will be on the same page, and there will be no challenges when one administrator creates a GPO and another tries to edit it.

Windows Vista Central Store

Windows Vista management stations have a trick up their sleeve. That is, they can use a "central store" for ADMX and ADML files. Recall that the ADMX files are the definitions themselves, and the ADML files are the language-specific files for each ADMX file.

The idea is that the central store lives on every Domain Controller. So, after the central store is created, your Windows Vista management station simply looks for it—every time it tries to create or edit a GPO—and it will automatically use the definitions contained within the ADMX files inside the central store.

This means you don't have to worry about running around to each of your 20 management stations to update them whenever new ADMX files come out. You simply plop them in the central store and you're done. You don't even have to tell the Windows Vista management stations you did anything; they'll just automatically look and use the latest definitions!

Here's the best part: it doesn't matter what kind of Domain Controllers you have. Doesn't matter if you have Windows 2000, Windows Server 2003, Longhorn Server or a mix of all three. It's the Windows Vista management station which is doing the work to look for the central store in the place upon the Domain Controller.

Wait, I'm going to stop here, and take a big deep breath and say it one more time. Because I know you're reading fast and want to get to the good stuff. So, say it out loud if you have to: **It doesn't matter if you have Windows 2000, Windows Server 2003, Longhorn Server, or a mix of all three. It's the Windows Vista management station which is doing the work to look for the central store in the place upon the Domain Controller.**

Got it? You don't have to "sell" your boss into upgrading the whole server back-end just to get this cool Central Store stuff.

So, let's read on and make it happen.

Creating the Central Store

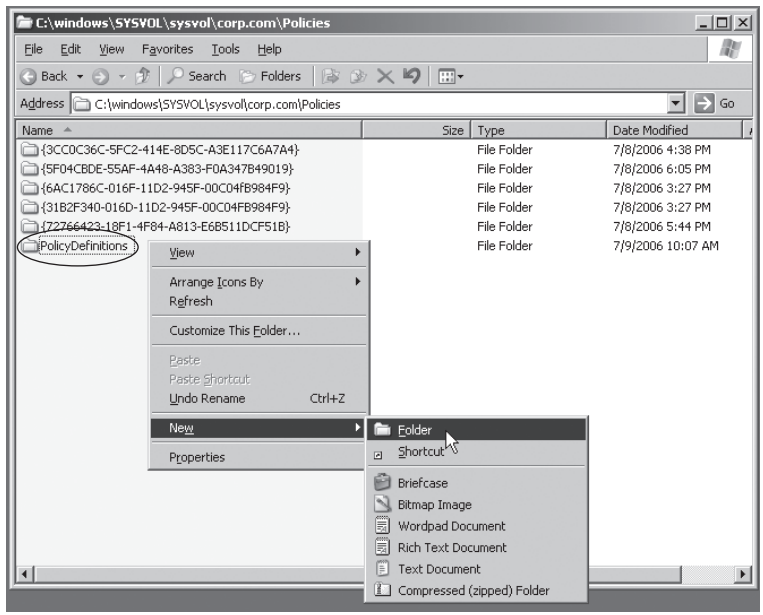
Creating the central store must be done by a Domain Administrator, because only a Domain Administrator has the ability to write to the location we need in SYSVOL. You can do this operation on any Domain Controller, because all Domain Controllers will automatically replicate the changes we do here to all other Domain Controllers via normal Active Directory/SYSVOL replication. However, it's likely best to perform this on the PDC Emulator, because that's the default location the GPMC and Group Policy Object Editor use by default.

To create the central store:

1. On the PDC emulator, use explorer or the command line to create a directory within `%systemroot%\sysvol\sysvol\`*`domain name`*`\policies`. You want to create a directory called `PolicyDefinitions` as seen in Figure 5.8.
2. We need a location to store our language-specific ADML files. Within `PolicyDefinitions` you'll create a directory for each locale. Again, U.S. English is US-EN. For other locales, visit `http://tinyurl.com/qpomo`.

Note that the directory name must be the same as specified in the locale reference page. If it's not, the ADMX file will not find its corresponding ADML file for that language.

FIGURE 5.8 Create a new directory called PolicyDefinitions in the Policies folder of SYSVOL



Populating the Central Store

Now, you simply have to get the ADMX and ADML files from your Windows Vista machine into the central store. There are a zillion possible ways to copy the files there. But, the steps are most easily done with two `xcopy` commands. This will work if your Windows Vista management station has access to the Domain Controller and that you have write rights.

To copy in the ADMX files into the central store from your Windows Vista management station:

```
xcopy %systemroot%\PolicyDefinitions\*
%logonserver%\sysvol\%userdnsdomain%\policies\PolicyDefinitions
```

To copy in the ADML files into, say, the U.S. English directory we created earlier:

```
xcopy %systemroot%\PolicyDefinitions\EN-US\*
%logonserver%\sysvol\%userdnsdomain%\policies\POlicyDefinitions\EN-US\
```

You can also use a free graphical utility for creating and populating the central store automatically at `www.gpoguy.com/tools.htm`.

Verifying You're Using the Central Store

Once you've created the central store directories in SYSVOL and copied the ADMX and ADML files to their proper location, you're ready to try it out! Start out by closing the GPMC if already open on your Windows Vista management station then re-opening it. You can fire up the GPMC by clicking Start and in the Run box typing **gpmc.msc**.

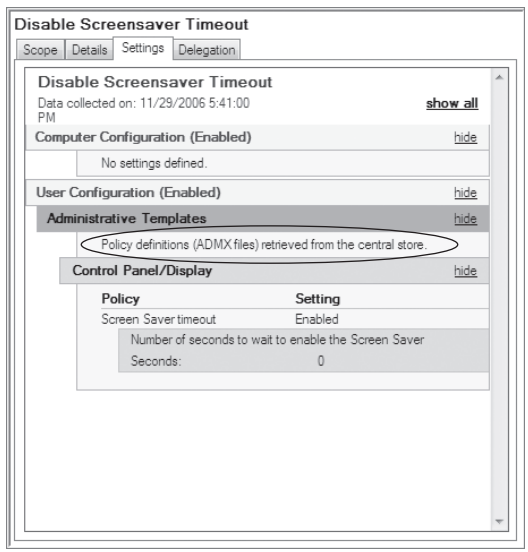
And, then just create and edit a GPO.

However, can you be sure you're really using the central store? Even if you miscopied the files or misnamed a directory, it would likely still look like it was working. That's because you'll simply "fall back" to using your local `c:\windows\PolicyDefinitions` directory which holds your ADMX files.

So, on one of your Windows Vista management stations, I suggest a little test. That is, rename the `PolicyDefinitions` directory within `c:\windows`. If you rename it to something like `zzz_PolicyDefinitions`, and you can continue to use the GPMC to create and edit GPOs—you therefore *must* be using the central store to pull those ADMX and ADML files.

There is a secondary test as well to help you verify that you're using the central store. That is, when you create and edit a GPO, then click the Settings tab inside the GPMC, you'll see a line under either Computer Configuration or User Configuration which says "Policy definitions (ADMX files) retrieved from the central store." You can see this in Figure 5.9.

FIGURE 5.9 Any time you click the Settings tab, the impromptu report will demonstrate if you are using the Central Store for your ADMX files.



Updating the Central Store

ADMX and ADML files will be updated. When Longhorn Server comes out, they'll be newer than the ones in the "out of the box" Windows Vista. Likewise, when Windows Vista's SP1, SP2, and so on comes out, those will be newer still, and so on.

When this happens, you'll need to update the central store, which couldn't be easier. Simply copy the latest and greatest ADMX files to the `PolicyDefinitions` directory you created in SYSVOL, and copy the latest and greatest ADML files to the language-specific directory within `PolicyDefinitions`.

Then you're done.

Additionally, other products, like Office 2007 will have ADMX and ADML files. If you wish to make those available to all administrators, just do the same thing. Drop them into the central store and you're done. (More about Office 2007 ADMX files a bit later.)

Office 2007 has, confusingly, both ADM templates and ADMX templates. As of this writing, ADMX templates are not available for download, but the ADM templates are. You can find the ADM templates here, `www.tinyurl.com/hzfcr`, but don't bother putting them in the Central Store, because ADM templates and the Central Store don't mix.

ADM and ADMX Templates from Other Sources

The templates Microsoft provides with Windows are just the beginning of possibilities when it comes to Administrative Templates. The idea behind additional templates is that you or third-party software vendors can create them to restrict or enhance features of either the operating system or applications.

If you know what to control, you're in business. Just code it up in an ADM or ADMX file and utilize it. If you're starting from scratch and have a choice, of course you'll want to use ADMX files instead of ADM files. That's because you can leverage the central store for ADMX files instead of remembering to copy ADM files to every management station.

However, it should be noted that you might already be using an ADM file or three. If you are, how do you get them to the ADMX "promised land"? A free tool, of course. Before we get into that, I will say that's the best option: get those custom and additional ADM files into ADMX format and leverage the central store. However, for completeness, I do want to explain what happens if you try to introduce an ADM file directly into a Windows Vista management station.

Using ADM Templates from Other Sources

Recall that ADM templates are the pre-Vista way to make definitions of what we can control. And, recall that there are both true *policies* and *preferences* which can be defined within an ADM file (or, ADMX file too).

Policies write to the "correct" place in the target computer's Registry. And, when the user or computer falls out of the "scope of management" of the GPO (that is, it doesn't apply to them anymore), the setting should revert back to the default.

Preferences write anywhere in the Registry that the application might be looking for it. Preferences tattoo the Registry. So, when the user or computer falls out of the scope of management of the GPO, the setting just sticks around.

You have the ability to get some ADM files from various sources. These ADM files sometimes have definitions for true policies. Other ADM files have definitions for preferences. How do you know which are which? The good news is, the Group Policy Object Editor interface shows you a difference between the two.

In Windows XP (and earlier) it shows blue for policies and red for preferences. In Windows Vista, it shows a little paper icon for Policies and a paper icon with a down arrow for preferences. That way, you can make an informed decision on whether or not you want to implement a preference.

Indeed, on GPanswers.com (`http://www.gpanswers.com/faq/` in the "Tips and Tricks" section halfway down the page), we have a gaggle of downloadable ADM templates that people have created to control various aspects of applications and of their systems.

Leveraging ADM Templates from Your Windows Vista Management Station

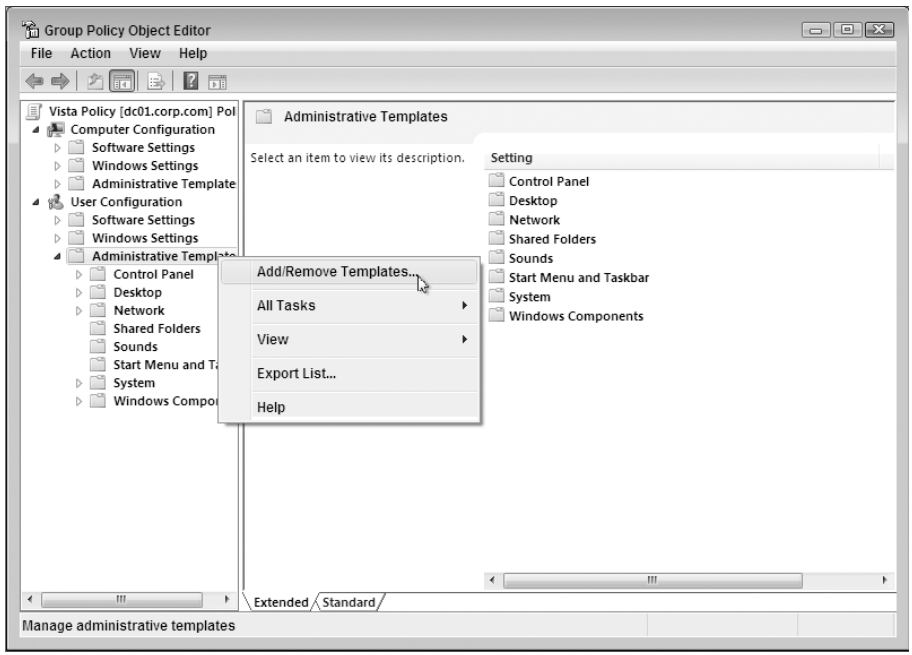
If you want to leverage and load one of these ADM templates into an existing GPO, simply edit it using the GPMC and bringing up the Group Policy Object Editor as seen in Figure 5.10. Then, choose either User Configuration ➢ Administrative Templates or Computer Configuration ➢ Administrative Templates, right-click over either instance of Administrative Templates, and choose Add/Remove Templates to open the Add/Remove Templates dialog box.

Click the Add button to open up the file requester, and select to load the ADM template you want. I'll show you in the next section or two where to track down more ADM files, but I wanted to show you this first so you'd know how to use them.

The original (pre-Vista) default location to start looking for ADM files is from `\windows\inf`; however, in practice you could store your ADM files anywhere. Just remember that every time you use an ADM template, you're copying that file directly into the GPO within SYSVOL.

When adding an Administrative Template, the interface suggests that you can choose to add it from either the Computer Configuration or the User Configuration node. In actuality, you can add the ADM template from either section, and the appropriate policy settings appear under whichever node the ADM template was designed for.

FIGURE 5.10 You can still Add/Remove Templates from a GPO you create with Windows Vista.



Once ADM templates are added using a Windows Vista management station, ADM templates show up under a special node within the Group Policy Object Editor, called "Classic Administrative Templates (ADM)" as seen in Figure 5.11. In Figure 5.11, I've loaded an ADM template for Word 2003 (again, I'll show you where to get these templates in a minute so you can experiment, too).

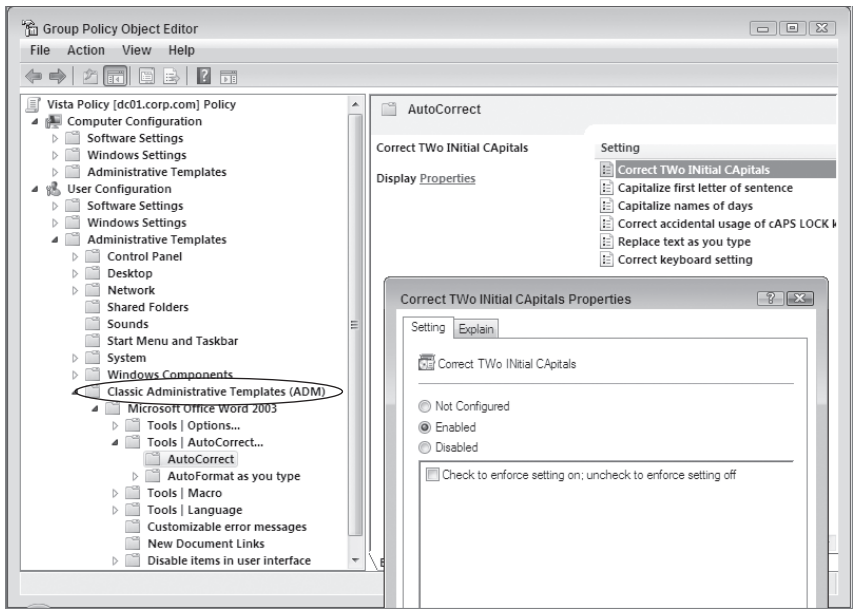
Viewing Old-Style Preferences

Again, ADM files can have definitions for true policies or for old-style preferences. If you load additional ADM templates into the Group Policy Object Editor (as shown in Figures 5.12), that contain old-style preferences, you simply won't see them as available in the Group Policy Object Editor.

A little later, I'll show you how to "create your own" ADM file which controls the Windows XP start sound. However, that trick won't produce a proper policy; rather, it produces an old-style preference.

Once you have the ADM file in hand, you'll load it into the GPO the same way. Choose either User Configuration ➢ Administrative Templates or Computer Configuration ➢ Administrative Templates, right-click over either instance of Administrative Templates, and choose Add/Remove Templates (see Figure 5.10) to open the Add/Remove Templates dialog box.

FIGURE 5.11 ADM templates are permitted in GPOs created from Windows Vista management stations. True policy settings are automatically available for use.



Click the Add button to open up the file requester, and select to load the ADM template you want. I'll show you in the next section or two where to track down more ADM files, but I wanted to show you this first so you'd know how to use them.

Once loaded, you'll see the same Classic Administrative Templates (ADM) node and the list of categories contained within the ADM file containing preferences (as seen in Figure 5.12). But, you won't see any settings. Again, that's because the Group Policy Object Editor automatically prevents you from seeing preferences—you need to turn on that ability inside the GPO.

To see the preference settings contained within the ADM file, use the menu at the top and select View ➢ Filtering to open the Filtering dialog box, as shown in Figure 5.13.

By default, the "Only show policy settings that can be fully managed" check box is checked. This is a safety mechanism that prevents old-style tattooing policies from being visible. Uncheck the check box, and you'll be in business.

After you turn on the ability to see the preferences within the interface, you'll notice that icons for old-style preferences have paper icon with a down arrow on them. This is to indicate that this is a preference and not a true policy, and these values will stick around even after the policy no longer applies to the user or computer.

Indeed, the Windows Vista Group Policy Object Editor is nice enough to even tell you this fact, as seen in Figure 5.14. You can see the little "down arrow" icon for any tattooing "preference."

FIGURE 5.12 ADM files containing preferences will show the categories available, but not the actual preferences until you enable that feature for the GPO.

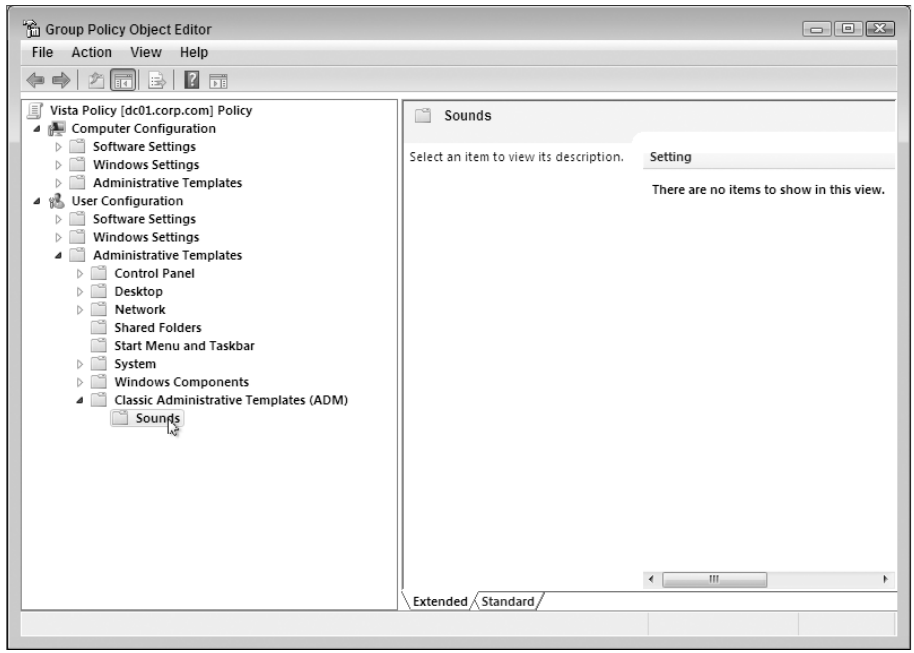


FIGURE 5.13 To see old-style preferences, clear the "Only show policy settings that can be fully managed" check box. This check box is checked by default to prevent you from seeing old-school preferences.

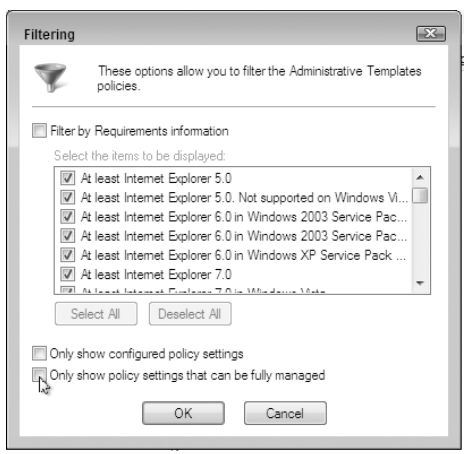
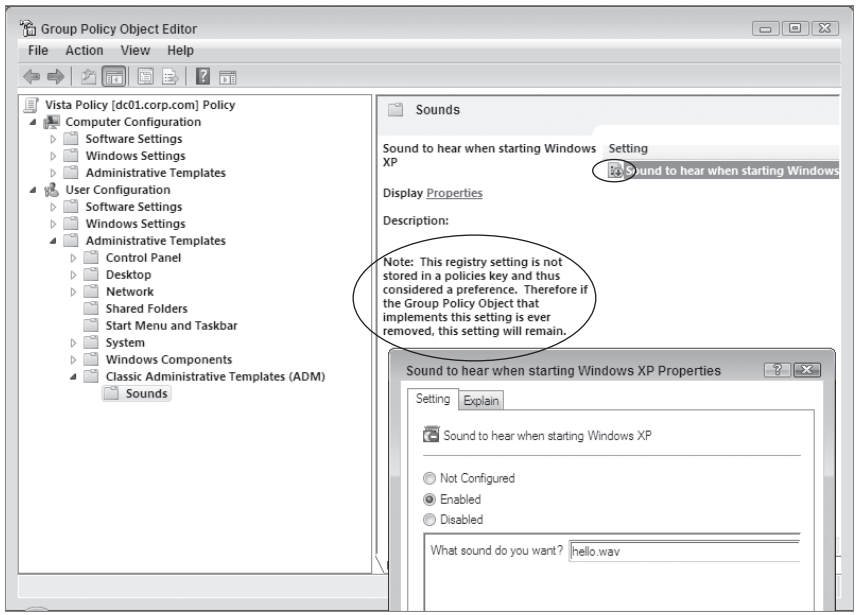


FIGURE 5.14 Windows Vista's Group Policy Object Editor warns you about the issues when leveraging preferences and not policies.



Microsoft Office ADM Templates

If you are also interested in deploying Office 2000, Office XP, or Office 2003, you'll be happy to know that they each come with a slew of customized ADM templates for you to import and use to your advantage.

- For Office 2000, download the Office 2000 Resource Kit tools at `www.microsoft.com/office/ork/2000/appndx/toolbox.htm`.
- For Office XP, download the Office XP Resource Kit tools at `www.microsoft.com/office/ork/xp/appndx/appa18.htm`.
- Office 2003 templates are located in the Office 2003 Resource Kit. Check `www.microsoft.com/office`. At last check, some even newer templates are available in the Office 2003/SP2 Resource Kit at `http://tinyurl.com/4wxxn`.
- Office 2007 templates are located at `http://tinyurl.com/2w9qs7`. However, Office 2007 is the first to also ship with ADMX templates (discussed a little later).

For information on how to automatically deploy Office 2000, XP, or 2003 (with patches and personalized customizations) to your users, see Chapter 10.

The file you're looking for (with either Office 2000 or XP) is called `Orktools.exe` (for Office 2003, it's `Ork.exe`), and it's about 9MB. After you install the corresponding Resource Kit on your management station, the following files are automatically placed in the `\windows\inf` folder for importation like the other ADM files.

Office 2000, Office XP, Office 2003, and Office 2007 ADM Templates

Here is a list of the ADM templates available for Office 2000, Office XP, Office 2003 and Office 207.

Office 2000 Templates	Office XP Templates	Office 2003 Templates	Office 2007 Templates	Description
access9.adm	access10.adm	access11.adm	access12.adm	Access settings
clipgal5.adm	gal10.adm	gaal11.adm	N/A	Restrict access to media clips
excel9.adm	excel10.adm	excel11.adm	excel12.adm	Excel settings
frontpg4.adm	fp10.adm	fp11.adm	N/A	FrontPage settings
instlr1.adm	instalr11.adm	instalr11.adm	N/A	Windows Installer settings
office9.adm	office10.adm	office11.adm	office12.adm	Common Office settings
outlk9.adm	outlk10.adm	outlk11.adm	outlk12.adm	Outlook 2000 settings
ppoint9.adm	ppt10.adm	ppt11.adm	ppt12.adm	PowerPoint settings
pub9.adm	pub10.adm	pub11.adm	pub12.adm	Publisher settings
word9.adm	word10.adm	word11.adm	word12.adm	Word settings
N/A	N/A	N/A	visio12.adm	Visio Settings
N/A	N/A	aer.adm	N/A	Corporate Windows Error Reporting (see the "Microsoft Corporate Error Reporting" section later in this chapter)

Office 2000 Templates	Office XP Templates	Office 2003 Templates	Office 2007 Templates	Description
N/A	N/A	`rm11.adm`	N/A	Microsoft Relationship Manager File location
N/A	N/A	`scrib11.adm`	`onent12.adm`	Microsoft OneNote 2003 settings
N/A	N/A	N/A	`cpao12.adm`	Calendar Printing Assistant for Outlook 2007
N/A	N/A	N/A	`groove12.adm`	Groove 2007
N/A	N/A	N/A	`ic12.adm`	Office InterConnect 2007
N/A	N/A	N/A	`inf12.adm`	InfoPath 2007
N/A	N/A	N/A	`proj12.adm`	Project 2007
N/A	N/A	N/A	`spd12.adm`	Sharepoint Designer 2007

Implementing a Customized Office Policy

After the Office templates are on the server, you can simply load them alongside the currently loaded templates. You can load all, some, or none—it's up to you.

In this example, we'll make believe we need to set up a custom Word 2000 policy for a collection of users. Normally, as in this example, Office template settings are meant for users, not computers. However, Office does include computer-side settings that you can use to override user-side settings if you want.

If you don't want to use the Office 2000 ADM templates in this example, you can substitute Office XP or Office 2003 templates. Just make sure you also have the corresponding Office suite installed on the target machine!

Here, you'll see how to use an additional template. We'll load the `WORD9.ADM` template alongside our current default templates. Then, we'll change the default behavior of our Human Resources users for Word 2000 as follows:

- The grammar checker is turned off while we type in Word.
- The spell checker is turned off while we type in Word.
- Word will ignore words in uppercase during spell check.
- Word will ignore words with numbers during spell check.

To change Word's default behavior for the **Human Resources Users** OU, follow these steps:

1. Log on to your Windows Vista management station as the Domain Administrator.
2. Download the Office 2000 Resource Kit tools and make sure the ADM templates are properly installed in the `\windows\inf` folder.
3. Fire up the GPMC.
4. Right-click **Human Resources Users** OU and select "Create and link a GPO here."
5. Create a new GPO called "Word 2000 Settings."
6. Edit the "Word 2000 Settings" GPO.
7. Choose either User Configuration ➢ Administrative Templates or Computer Configuration ➢ Administrative Templates, right-click over either instance of Administrative Templates, and choose Add/Remove Templates to open the Add/Remove Templates dialog box.

When adding an Administrative Template, the interface suggests that you can choose to add it from either the Computer Configuration or the User Configuration node. In actuality, you can add the ADM template from either section, and the appropriate policy settings appear under whichever node the ADM template was designed for.

8. Click the Add button to open up the file requester, and select to load the `Word9.adm` template from the `\windows\inf` folder. Click Close to close the Add/Remove Templates dialog box to return to the Group Policy Object Editor.
9. To turn off the "Check Grammar As You Type" feature, drill down to User Configuration ➢ Administrative Templates ➢ Microsoft Word 2000 ➢ Tools ➢Options ➢ Spelling & Grammar ➢ Check Grammar As You Type. Then, enable the setting, but do *not* select the check box. This forces the policy on the user, but clearing the check box forces it off.
10. Repeat step 9 for "Check Spelling As You Type," "Ignore Words in Upper case," and "Ignore Words with Numbers."

You can try this exercise with the other Office 2000–supplied templates listed earlier. These will affect Excel, PowerPoint, Access, and the like.

To test your new policy on the **Human Resources Users** OU, simply log on to any Windows 2000 or Windows XP machine loaded with Word 2000 as a user who would be affected by the new policy. For instance, log on to XPPRO1 as Frank Rizzo, our old HR pal from Chapter 1 (assuming you have Word 2000 loaded).

Then in Word, choose Tools ➢ Options to open the Options dialog box, and make sure the settings reflect the policy settings you dictated.

Now, in this example we just explored, we were using the raw ADM files. Again, you can (as you'll discover a little later) take these ADM files and covert them—lock, stock, and barrel, into ADMX files to be used in the central store.

Also note that Office 2007 will have available downloadable ADMX files—no need to convert or do anything fancy. Just plop 'em in your Central Store and start using them. We'll talk more about the Office 2007 ADMX files a little later. Check it out in the upcoming section "Using ADMX Templates from Other Sources."

Other Microsoft ADM Templates

Microsoft has two additional applications outside the Office family of products that leverage the Group Policy infrastructure by using ADM templates.

Microsoft Software Update Services (SUS) and Windows Server Update Services (WSUS)

The job of Microsoft's Software Update Services (SUS) and the newer Windows Server Update Services (WSUS) is to ensure that patches are deployed to your Windows 2000, Windows XP, and Windows 2003 client systems. After a server is set up to deploy the patches, the client system learns about the server by way of a custom ADM template.

The template is built in to Windows 2003 and Windows 2000 + SP4 as `Wuau.adm`. However, the template is not built in to Windows 2000 + SP3.

You can learn more about SUS, how to deploy it, and how to use the rather complex ADM templates from two articles I wrote for *MCP Magazine*, which you can find at `http://tinyurl.com/86sbj` and `http://tinyurl.com/5gfuk`. These articles form a two-part series about installation and troubleshooting. The latter's main focus is on understanding the ADM template. Lastly, Microsoft has an excellent guide to the policy settings with regard to WSUS available at `http://tinyurl.com/8nalu`.

Microsoft Corporate Error Reporting

Microsoft has a service that lets corporate IT administrators "trap" error messages to a central server, instead of being sent directly to Microsoft, which is called Corporate Error Reporting (CER). CER can help track systems that frequently crash and can provide an easier way to connect with Microsoft if a system does fail often. It can trap information for a lot of Microsoft's most popular applications including Office XP, Windows XP, Windows 2003, Project 2002, and Sharepoint Portal Server.

Microsoft CER uses the ADM file `Cer2.adm`. You can get more information on CER at `www.microsoft.com/resources/satech/cer/`. You'll find the ADM file in the "toolbox" section of the web page.

ADM Templates You Shouldn't Use with Windows 2000, Windows XP, or 2003

Both the Office 2000 Resource Kit and Windows 2003 Server itself come with additional ADM templates that are not truly meant for the Group Policy Object Editor. Make a note of them so that you don't use them by mistake.

Office 2000 NT/95 Templates

Additional settings to configure Internet Explorer 5 are included in the Office 2000 Resource Kit, but they are not automatically copied to the `\windows\inf` folder. These are found, after the Office 2000 Resource Kit is installed, in the `\Program Files\IEAK\policies\EN` folder. The ADM policies in the ADM templates (located in the following table) are *not* meant for the Group Policy Object Editor. Rather, these are for the old-style Windows NT/95 `Poledit.exe` program.

Internet Explorer 5 Templates	Description
`Aaxa.adm`	Data binding settings.
`Chat.adm`	Microsoft Chat settings.
`Conf.adm`	NetMeeting settings.
`Inetcorp.adm`	Dial-up, language and temporary Internet files settings.
`Inetres.adm`	Internet properties, including connections, toolbars, and toolbar settings. Equivalent to the Tools ➢ Internet Options command.
`Inetset.adm`	Additional Internet properties: AutoComplete, display, and some advanced settings.
`Oe.adm`	Outlook Express Identity Manager settings. Use this to prevent users from changing or configuring identities.
`Sp1shell.adm`	Active Desktop settings.
`Subs.adm`	Offline Pages settings.

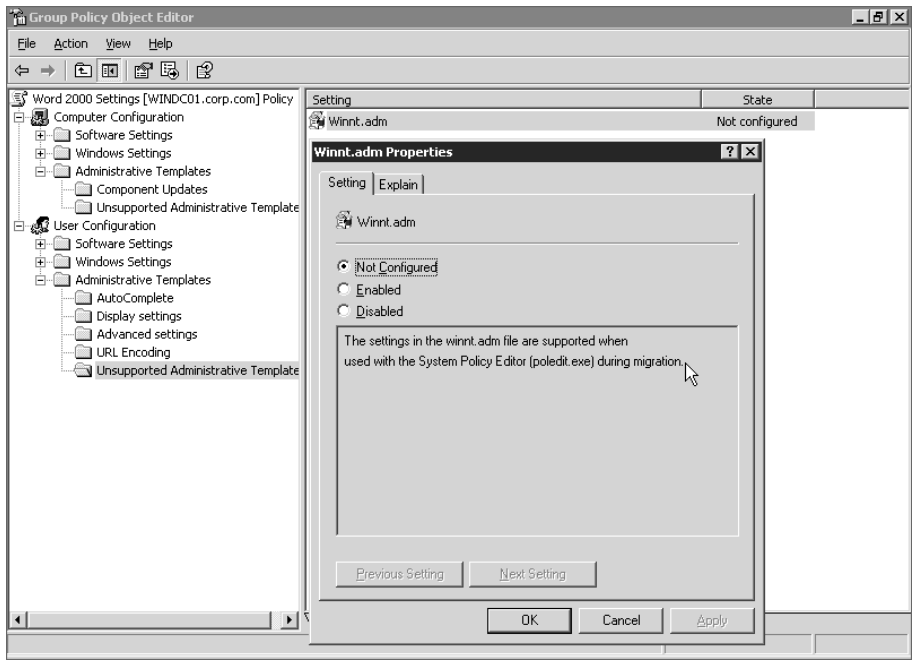
Some of these templates *can* be loaded into Windows 2000, but you probably wouldn't want to do so; some settings included in these templates include actual policies (nontattooing), and some include only preferences (only tattoo). To review the difference between policies and preferences, see the opening section of this chapter. You can just use the included Internet Explorer template settings found in Windows 2000's `inetres.adm`, instead of loading these templates that include both policies and preferences.

Windows NT Templates

Additionally included with a Windows 2000 computer are even more ADM templates. These are not for use within the Windows 2000 Group Policy Object Editor either; rather, they are for use with the old-style NT `Poledit.exe` program. This feature set includes the following:

Windows NT Template	Function
`Common.adm`	User interface options common to Windows NT 4 and Windows 9*x*. For use with System Policy Object Editor (`Poledit.exe`).
`Inetcorp.adm`	Dial-up, language, and temporary Internet files settings. For use with System Policy Object Editor (`Poledit.exe`).
`Inetset.adm`	Additional Internet properties: AutoComplete, display, and some advanced settings. For use with System Policy Object Editor (`Poledit.exe`).
`Windows.adm`	User interface options specific to Windows 95 and Windows 98. For use with System Policy Object Editor (`Poledit.exe`).
`Winnt.adm`	User interface options specific to Windows NT 4. For use with System Policy Object Editor (`Poledit.exe`).

These templates are really not 100 percent compatible with the Group Policy Administrative Template interface if imported directly. Some will indicate that they are unsupported, as shown here.



These are to be used with the old System Policy Object Editor (`Poledit.exe`). For instance, if you do end up loading, say, the `Winnt.adm` into the Windows 2003 Group Policy Object Editor, you are informed that it won't work, and the settings will not be displayed.

Using ADMX Templates from Other Sources

You'll get ADMX files the same way you got ADM files: companies like Microsoft will make them available to control the products they support, and enterprising geeks will produce ADMX files which control other parts of the operating system and third-party applications.

The same basic note and warning applies though: ADMX files can contain both (or either) true policies or old-school preferences. And, if they do contain preferences, you'll need to explicitly show them in the Group Policy Object Editor as seen in Figure 5.14.

ADMX Templates for Office 2007

As of right now, Office 2007 has been released, but the ADMX files for it have not been released.

But, when they are, you already know what to do. Just chuck 'em in the Central Store (both ADMX and ADML files in the appropriate places) and you'll be golden. Then all the new GPOs that you create will be able to control Office 2007!

ADMX Templates from Other Sources

Will other Microsoft products have ADMX files? We hope so. So, while I have nothing specific to report now, check in every so often on GPanswers.com (especially the newsletters, where I'll try to let you know about any new ones that pop up).

Darren Mar-Elia, who runs GPOguy.com and is the technical editor of this book's edition has an ADMX version of his troubleshooting tool, called GPOLOG.adm at www.gpoguy.com/gpolog.htm.

Deciding How to Use ADMX Templates

Once you have the ADMX templates, you need to decide how to use them. If you've already created the central store—terrific. Just plop them into the central store and you're done. However, note that this means that all administrators who have access to create GPOs using Windows Vista management stations will be able to leverage this ADMX file.

You might not want to enable all administrators to leverage this ADMX template.

If that's the case, you only have one option: put the ADMX files you want to use on the Windows Vista management station you use. The downside, however, is that if another Group Policy administrator (on his Windows Vista management station) tries to edit the GPO or report on it, he won't get the same view of all the settings that you do. That's because his Windows Vista management station doesn't contain the ADMX file you're using.

So, best practice is to use the ADMX file central store whenever possible.

ADMX Migrator and ADMX Editor Tools

Since leveraging ADM files directly inside GPOs which also use ADMX files can be fraught with peril, wouldn't it be a better idea to just utilize ADMX files everywhere? That way, you can just plop 'em all in the central store and be done. If you already have custom ADM files and need to get them to ADMX land, there's a free utility which was written by FullArmor Corporation and licensed by Microsoft to give to you for free.

It's got a silly name: the ADMX Migrator tool. Doesn't it sound like it migrates ADMX files? Well, it doesn't. Maybe it should have been called ADM2ADMX or something, but, regardless of the name, it's a cool tool. You can download the tool from Microsoft here: http://tinyurl.com/ydb6ub. (Believe me when I tell you the non-Tiny-Url would choke a horse.) Note that it first requires the .NET Framework 2.0 to be previously installed.

Additionally, inside the ADMX Migrator tool package is a neat ADMX editor as well to help you hand-craft your own ADMX files from scratch. The idea is that you don't have to "learn" a new language and hand-code it using, say, Notepad. Just use the tool to create your own ADMX files and you're in business.

For these examples, I'm running the tools on my Windows Vista management station, but they'll work just fine on a Windows XP which has the .NET Framework 2.0 loaded as well.

ADMX Migrator

There are lots of places you can get pre-made ADM files. You might try leveraging some right now—some are at GPanswers.com, others are found online from various other websites. Here's an example of a simple ADM file if you want to follow along. Just take this text, and copy it into Notepad and save it as Sounds.ADM.

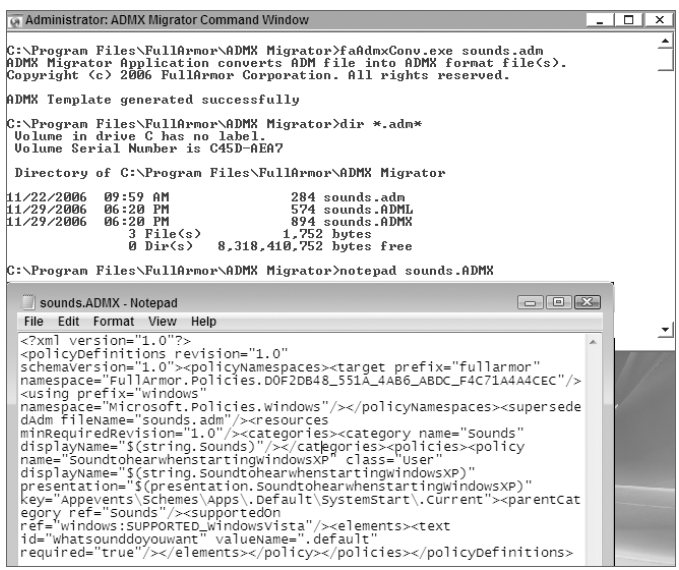
```
CLASS USER

CATEGORY "Sounds"
   POLICY "Sound to hear when starting Windows XP"
      KEYNAME "Appevents\Schemes\Apps\.Default\SystemStart\.Current"
      PART "What sound do you want?" EDITTEXT REQUIRED
      VALUENAME ".default"
      END PART
   END POLICY

END CATEGORY
```

Then run the tool named faAdmxConv.exe against the ADM file you have. It can be as simple as just pointing to the file, but there are more switches if you have specific requirements. Once run, it will create an ADMX and ADML file for the ADM, which is then ready to be put in the central store (or, if you're not using the central store, then with individual Windows Vista management stations). You can see the program run and its output in Figure 5.15.

FIGURE 5.15 The faAdmxConv.exe tool will take your ADM and convert it into an ADMX and ADML file.



Then, if you want to leverage these in the central store, put the ADMX file in the `\PolicyDefinitions` directory within the SYSVOL and the ADML file in the language directory (en-US for English).

ADMX Editor

In the previous example, we leveraged an existing ADM file which modified Windows XP's startup sound. What if you wanted to create the ADMX file from scratch?

Creating an ADMX template can sometimes be difficult. The hardest part can be figuring out which Registry setting you need to modify on the client system. You can use several tools to help you. One such tool is ProcessMonitor from Microsoft's Sysinternals tools. You can find it at `http://tinyurl.com/y45pu7`. This tool can help point out what's changing on the client.

Then, armed with that information, you can triumphantly create your own custom ADM or ADMX template and try it. That's where the ADMX editor, also in the ADMX Migrator download comes in.

Once you fire it up, you'll be able to create a new ADMX file and add categories, like "Misc XP Sounds" as seen in Figure 5.16. Note that it's not easy (at all) to realize you need to click to the right of Display Name to get that field to turn on. Once you do, you can enter in the name.

Then, right-click over your new category and enter in your first policy setting. Here, we're only entering one: "Sound to hear when starting Windows XP." We then give it the Registry key (seen in the previous ADM listing) and the Registry value name (also seen in the previous ADM listing) and finally specify that it's a User-side setting with the pull-down menu next to Class. You can see these all entered in Figure 5.17.

Then, you can add different "Elements," such as a Dropdown List, ComboBox, and more as seen in Figure 5.18. You can also enter in your own Explaintext and Supported On text.

FIGURE 5.16 Once you create a new ADMX file, you can create your first category, such as "Misc XP Sounds."

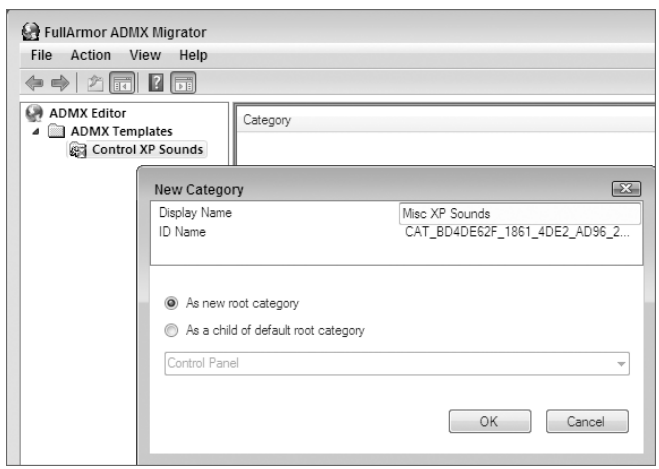


FIGURE 5.17 You can create your own policy settings within the categories you previously created

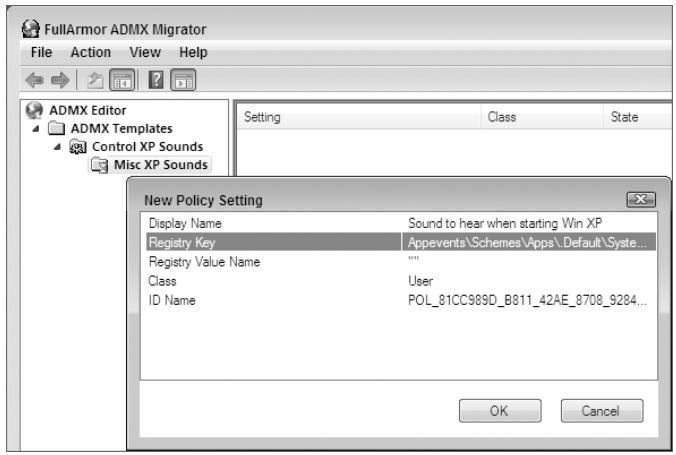
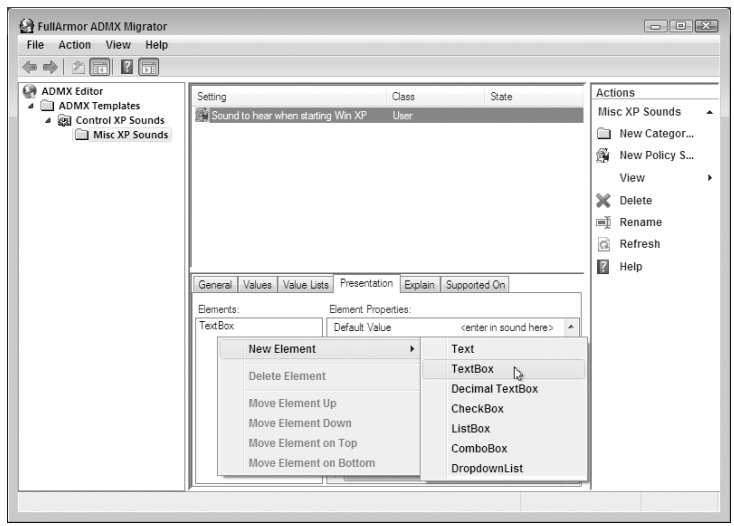


FIGURE 5.18 You can add various elements like TextBox requesters, DropdownLists and more.



When ready, you can right-click over the ADMX file (in my example the node labeled "Control XP Sounds") and click Save As. This will create an ADMX and ADML file. Be sure to (again) move the ADMX into the central store and the ADML file into the language directory (en-US) for English.

I really wish there was some kind of "preview mode" to see if you got it right before you went through the motion of copying the ADMX and ADML files to their final location. Because there's potentially a lot of trial and error involved before you get it just right.

When you do get it right, however, and fire up the GPO editor you'll notice that the category is there (in my example it's called Misc XP Sounds), but the settings within it are absent. That's because the keys we're dictating aren't part of the proper Policies keys, and hence won't show by default. If you want to expose them, you'll need to select View ➢ Filtering from the Group Policy Object Editor window, and in the Filtering dialog, *uncheck* "Only show policy settings that can be fully managed." When you do this, you'll see the setting show up, with a little down-arrow designating that it's not a true Policy setting, as seen in Figure 5.14.

Finding the Policy Settings You Need and Cracking the ADM/ADMX Files

I get about ten e-mails a day which ask me, "Hey Jeremy, how do you 'X' with Group Policy?" (where X is some policy or trick I've never personally tried to do before).

My standard answer is: "I don't know" because I simply don't have all 1800 Windows XP and certainly not all 2400 Windows Vista policy settings memorized. So, I immediately follow up my "I don't know" with "But we can find out!"

Microsoft's *Policy Settings Spreadsheets* for Windows XP and Windows Vista

Microsoft has created two wonderful documents: one for Windows XP and one for Windows Vista.

In short, you can download a Microsoft Excel spreadsheet detailing the following:

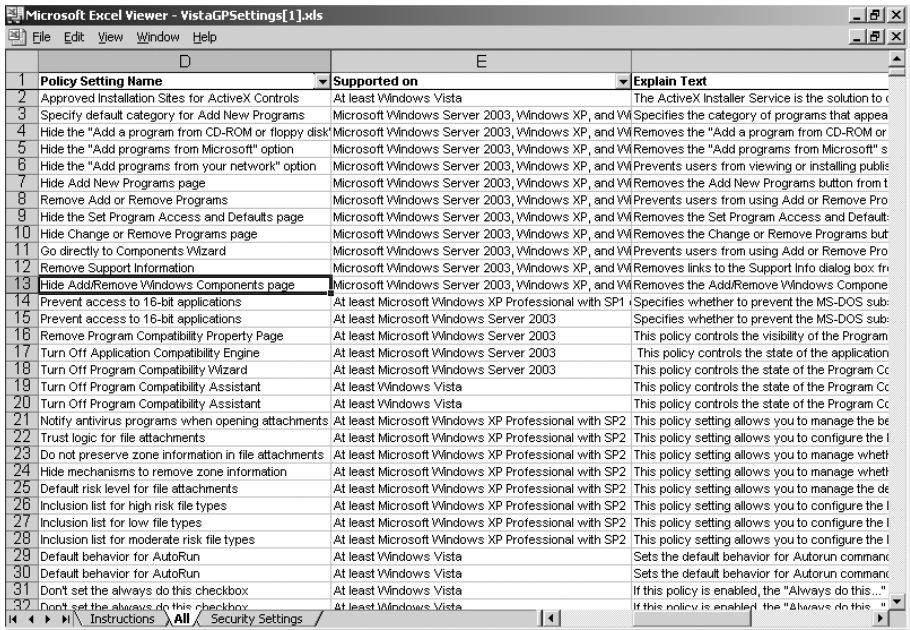
- Every policy setting
- Every path to every policy setting (User or Computer, Administrative Templates ➢ etc.)
- Every security setting
- Every Explaintext entry for each policy setting
- Every Registry punch for every policy setting

I always keep an updated pointer from my GPanswers.com website in the Microsoft resources to this spreadsheet. Again, at this time, Microsoft has one for Windows XP management stations and one for Windows Vista management stations. I don't know how much longer they plan on keeping the Windows XP version around. You can see what this looks like in Figure 5.19.

Additionally, since it's just Excel, you can perform quick sorts. For instance, by using column E (Supported on), you can limit the view to show you, say, only Windows XP+SP2 settings.

This is super handy.

FIGURE 5.19 The PolicySettings.xls settings reference spreadsheet



Last Ditch Effort Troubleshooting via Registry Punch

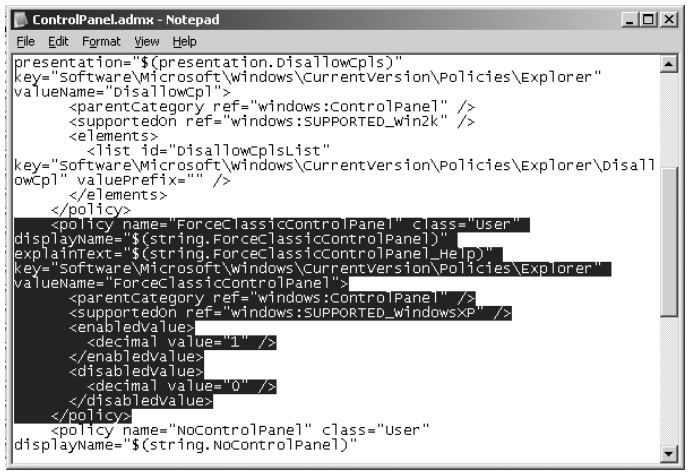
Chapter 4 discussed many ways to troubleshoot if Group Policy doesn't seem to be applying. However, if you've verified that you're getting the policy setting via GPRESULT or the Group Policy Results Wizard and you're certain you should be getting a specific policy you set, perhaps the problem is elsewhere.

Occasionally, there are bugs in the help text definitions of some policy settings in the ADM/ADMX templates. Sometimes the policy setting states that "Enable" does one thing and "Disable" does another—and, really, it doesn't work that way at all. Other times, the actual underlying definition of the policy setting is incorrect, and the Registry location it's set to modify doesn't really do anything. In all honesty, these problems are few and far between, but it is precisely what service pack updates to existing ADM files try to correct.

So, if you're 1000 percent convinced you're getting the GPO laid down on the client system, yet you're still not seeing the result of a specific policy setting, take it to the next step. That is, crack open the spreadsheet (or ADM/ADMX file itself), locate the policy definition, find the portion of the Registry that the policy will be setting, and manually enter that hack into your client system. After you do, verify it against what the policy setting says it's supposed to do.

Is it actually doing what it says it's supposed to do? For instance, if you suspected that the **Force Classic Control Panel View** policy setting wasn't doing what it says it was going to do, simply crack open `ControlPanel.admx` using Notepad, and locate the **Do not display Manage Your Server Page** policy setting (as shown in Figure 5.20).

FIGURE 5.20 Open the ADMX template to locate the policy and the corresponding Registry hack.



As you can see, the policy setting modifies `Software\Microsoft\Windows\CurrentVersion\Policies\Explorer`. It adds a value of `ForceClassicControlPanel` and sets it to 1 to force the XP Control Panel to revert back to the older Windows 2000 style.

You can plunk this into the Registry yourself and see this actually happen; you don't need to set a GPO to try it. After you verify the results, you're closer to knowing precisely what's going on.

Final Thoughts

Managing ADM and ADMX files can be a little tricky. The key message to take away is always use a Windows Vista management station to do your editing. If you bounce around using various operating system types, you'll be back in "SYSVOL bloat" hell again.

It's easy to use Microsoft and third-party, vendor-supplied ADM templates to control your applications or to make your own ADM modifications. But remember—only applications coded to read Registry settings from the Policy keys will be true Policies. They will be applied and removed when different users log on or off. They will not tattoo. They will appear with a paper icon (in Windows Vista) or a blue dot (in pre-Windows Vista versions) in the Group Policy Object Editor. Most applications are not yet Policy key–aware, which means if you want to create your own modifications, you'll likely need to make them preferences. Preferences do not modify the Policy keys. They tattoo the Registry. They will appear with a down-arrow (in Windows Vista) or a red dot (in pre-Windows Vista versions) in the Group Policy Object Editor.

If you want an application which can truly policy-enable your existing applications, check out PolicyPak.com.

Be wary of download ADM templates you find online. They'll usually work as advertised, but the problem, again, is that they're likely chock full of irritating tattooing preferences, not lovely nontattooing policies. One site that's full of such ADM templates is `http://worldofasp.com/ts/download.cfm`. Of course, I have some free ADM templates to download at `www.GPanswers.com/faq`.

If you have an ADM file you want to use in the central store, you'll have to convert it to ADMX first. Use the downloadable ADMX Migrator tool to perform that magic. Additionally, use the ADMX Editor (part of the ADMX Migrator download) to hand-create your own ADMX files if you like.

If you're interested in hand-creating ADMX files, we will have tips and tricks and a forum on `GPanswers.com`. We will also maintain the previous edition's "ADM Template Syntax" section if you need that as well. Lastly, check out Microsoft's document to "Step-by-Step Guide to Managing Group Policy ADMX Files" at `http://go.microsoft.com/fwlink/?LinkId=55414`. And for the truly geeky, you can check out the ADMX schema, located at `http://tinyurl.com/28k56v`.

Practical Guidance with Detailed Coverage of Windows Vista, XP, and Server 2003

This revision of the popular *Group Policy, Profiles, and IntelliMirror* is fully updated for Windows Vista. Inside, you'll learn how best to use Group Policy to take full advantage of Active Directory and create a managed desktop environment. You'll learn details about the GPMC, Group Policy troubleshooting techniques, and configuring Group Policy to create a resilient desktop environment. Inside, discover how to:

- Master all major Group Policy functions for all versions of Windows
- Troubleshoot Group Policy using tools, logs, Resource Kit utilities, registry hacks, and third-party tools
- Use Group Policy to secure your Windows Vista and Windows XP desktops
- Create and manage ADMX files and leverage the Group Policy Central Store
- Deploy Office 2007, Office 2003, and more using Group Policy Software Installation
- Utilize Windows Deployment Services to roll out new desktops
- Script complex Group Policy operations, including linking, backup, restore, permissions changes and migrating
- Set up roaming and managed desktops between XP and Vista machines
- Control hardware, restrict software, assign printers, and tweak Internet Explorer 7

www.sybex.com

Jeremy Moskowitz, founder of Moskowitz, inc. (www.moskowitz-inc.com), is an MCSE, MCSA and one of only six Microsoft MVPs (Most Valued Professionals) in Group Policy. He has performed Active Directory, Group Policy and Windows management consulting for some of the nation's largest organizations. Jeremy teaches his very popular Group Policy Intensive Training and Workshop classes and runs **GPanswers.com**, a community portal to help answer tough Group Policy questions. Jeremy frequently contributes to *Windows IT Pro Magazine*, *REDMOND* Magazine, and *Microsoft Technet Magazine,* and he has been a noted speaker at many industry conferences.

Coverage Includes: ADMX files, the Central Store, Offline Files Updates , Windows Deployment Services, and More

Master Advanced Features: Inheritance Blocking, Prioritization, Linking, Loopback Policy Processing, Security Policy Processing, Enforcing, and WMI Filters

Go Beyond Group Policy: Deploy Shadow Copies and Utilize Windows Deployment Services for Complete Protection of Your Data and Systems

About the Series

The Mark Minasi Windows Administrator Library equips system administrators with in-depth technical solutions to the many challenges associated with administering Windows in an enterprise setting. Series editor Mark Minasi, a leading Windows expert, not only selects the topics and authors, he also develops each book to meet the specific needs and goals of systems administrators, MIS professionals, help-desk personnel, and corporate programmers.

ISBN: 978-0-470-10642-6

$49.99 US
$59.99 CAN
£31.99 UK

Sybex®
An Imprint of
WILEY

BICENTENNIAL 1807 WILEY 2007 BICENTENNIAL

COMPUTERS/Networking/
General Operating Systems/Windows