

Designing and Implementing Group Policy in the Enterprise

In the first three chapters of this book, I gave you the foundation for creating and managing Group Policy Objects (GPOs) in your Active Directory (AD) infrastructure. But to realize the benefits of GPOs—benefits such as reducing the costs of managing a Windows 2000 (Win2K) infrastructure, you need to be able to apply what you’ve learned so far to a real-world environment. My goal in this chapter is to build on the previous chapters by giving you practical advice for deploying GPOs in a large-scale Win2K infrastructure. Indeed, the capabilities and complexity of GPOs don’t become evident until you begin to design and deploy group policy in a real-world situation.

The challenges of designing a GPO implementation correctly have resulted in a relatively slow rate of deployment in the enterprise. If you look at most large enterprises that have deployed Win2K, a small percentage have made the jump to an AD implementation, and an even smaller percentage have committed to a robust deployment of GPOs.

In this chapter, I’ll describe my experiences in designing such a deployment and provide best practices for deploying this powerful technology. The exciting thing about GPOs is that they have the potential to give you unprecedented control over your Windows infrastructure. The challenge is that they require a lot of careful planning and even more careful management to fulfill this promise.

Integrating Group Policy Design in Your AD Infrastructure

The first step in deploying group policy is planning a design that works with your AD infrastructure. In fact, I recommend having a good idea about how you want to use GPOs during your AD planning process. If you think about how GPOs are structured and the fact that they’re linked to sites, domains, and Organizational Units (OUs), you can see why a particular AD design might make it more or less difficult to deploy a viable GPO design. As I discussed in Chapter 3 of this book, the goal of your AD design is to meet your business needs as well as those of your technology.

For example, if your company is organized into divisions (Engineering, Finance, Sales, and so on), you might be inclined to create an OU design that mimics that structure. However, if you also plan to use GPOs to manage the resources in the company, think about how a GPO implementation might map into such a structure. And, perhaps just as important, who’ll manage the GPOs that you design, and how much latitude will you grant them to make changes or create new ones?

In fact, this last point often goes to the very heart of integrating a GPO design with an AD design. Remember that AD provides very granular delegation for administering objects. Your Engineering OU might be managed centrally by your corporate system administrators, or management might be delegated to someone in the Engineering division who is responsible for just those objects. In either case, you’ll need to think about where you’ll link GPOs and who’ll manage them.

Remember that GPOs are inherited. So if you link one GPO at the domain level and another at the Engineering OU level, both GPOs can affect the users and computers in that Engineering OU. And, as I discussed in earlier chapters, you can filter the effects of policy using security

groups. Thus, you might be trying to decide whether to link most of your GPOs at the domain level, letting security-group filtering control where they apply, or to link more selectively at the OU level for a given set of users and computers.

To complicate matters even more, GPOs are monolithic. That is, a particular GPO, by default, can contain policy for everything from security to Internet Explorer (IE) lockdown to Administrative Template lockdown. You can work around this to some degree by not allowing certain GPO administrators to load certain Microsoft Management Console (MMC) snap-ins related to Group Policy. (See “Scenario B—Multiple AD Domains” later in this chapter.) But in the end, you may opt to create many GPOs, each of which provides single (or few) policy functions as opposed to fewer GPOs that do everything. Lastly, any GPO design needs to take GPO processing performance into consideration. A situation in which a computer or user has to process tens or hundreds of GPOs that may be housed in more than one domain might result in an unacceptable user experience.

By now, you’re probably asking yourself why in the world anyone would deploy GPOs and how anyone can reap the expected benefits from them in the face of this complexity. I know I’ve asked this question myself on several occasions! Let me assure you that you *can* design and deploy a GPO implementation that maximizes the incredible control this feature provides while minimizing the management complexity of the resulting infrastructure.

What I’ve talked about so far in this chapter can be distilled into the following five questions, which you need to consider as you’re designing your GPO implementation:

1. Does your AD design take into consideration how you’ll use GPOs?
2. What GPO functionality do you want to use in your environment?
3. Will you create and manage AD objects (and thus GPOs) from a central group in your organization, or will management be delegated across many business units? (It’s possible and perfectly reasonable that GPO delegation won’t match your planned delegation of AD objects.)
4. Will you create GPOs and link them right to the container where they apply (say, at an OU)? Or will you link them at a higher level (such as at the domain level) and use security-group filtering to control their effect?
5. How will you manage aspects of GPO administration such as change control, Resultant Set of Policy (RSOP), and GPO processing performance, and will the GPO delegation design you’ve created allow such management?

My goal for the rest of this chapter is to help you answer these questions and to discuss some of the technical issues you might encounter as you design your own GPO deployment.

Managing the Complexity of GPOs

Anyone who has designed infrastructure deployments of Windows NT or Win2K knows that Microsoft imposes—albeit unknowingly—a perpetual trade-off. On the one hand, you can reap the benefit of the lowered cost of ownership of advanced technologies like AD and GPOs, but you run the risk of negating that benefit by trying to manage their complexity. The reality for most enterprises is that they’re generally not in the business of running a Windows

infrastructure. They're in the business of running a business, and the Windows infrastructure is a tool to get that core job done. So, whenever you can use technology as an aid rather than a hindrance, life is good.

GPOs fall squarely into this category. Use them incorrectly, and they'll cost you a lot more money than they save. Use them well, and you're a hero, and you can truly claim that you're taking full advantage of the infrastructure and saving your company money. I've found that the best way to take full advantage of any complex technology is to use that technology in baby steps. Unless you're completely comfortable with all aspects of your Win2K infrastructure and its management, there's no reason to deploy all of the features of GPOs all at once. Start small and start simple!

Your initial GPO design should seek to do very little. As I pointed out in chapters 1 and 2, the minimum you'll need to do is use GPOs to set security policy on your domain. If you've been using system policies in NT 4.0, you might decide to duplicate that desktop lockdown functionality using Administrative Template policy within a GPO or two. If you're using the Internet Explorer Administration Kit (IEAK) to lock down your users' IE configurations, you might add some IE Maintenance policies to your GPO. As I said, the idea is to start with aspects of GPOs that you're familiar with.

When you're deciding where to link the GPOs, again—start simple by linking them to the domain and having them apply to all users and computers. If you don't want a policy setting applying to everyone in a particular domain, link it to an OU, but don't go overboard. Finally, when you begin to deploy your GPO implementation, don't immediately delegate control of the GPOs to every administrator on your network. Keep them centrally managed by a group of folks who are familiar with the technology. This gives you time to refine your GPO change control, management, and deployment processes before rolling out a larger deployment.

The following points summarize these “keep-it-simple” tips for your initial GPO deployment:

- Start with features in GPOs that you're already familiar with from NT 4.0 (for example, Administrative Templates and IE Maintenance).
- Initially, link GPOs to a few containers (such as domain or OU) to get a feel for managing and controlling them.
- Only deploy a couple of GPOs to begin with. Don't go all out and deploy dozens of GPOs at all levels of your AD infrastructure.
- Maintain centralized control and management of GPOs during their initial deployment phases until you understand GPO delegation and can put good change-control processes in place.

Once you've mastered the complexity of GPOs, you can experiment with more complex implementations or a larger number of GPOs. In the next section, I'll present some approaches to GPO deployment that you can use in your own AD infrastructure.

Designing the Implementation

How does one go about actually designing a GPO implementation? Earlier in this chapter (see “Integrating the Design into Your AD Infrastructure”), I presented five questions that you need

to ask yourself as you're designing your implementation. Now I'll present two scenarios based on different possible answers to those questions. Scenario A describes a fairly simple, single-domain implementation of GPOs, showing how to migrate from NT 4.0 system policies to a GPO environment. Scenario B is more complex; it considers multiple AD domains and the various ways you can design and deploy GPOs in such an environment.

Scenario A—One AD Domain

In this scenario, Company A is deploying a single AD domain, migrating from its current NT 4.0 infrastructure, and wants to start using GPOs slowly. Table 4.1 shows how Company A answers the five questions I presented earlier in this chapter (see "Integrating the Design into Your AD Infrastructure").

Question	Answer
Does your AD design take into consideration how you'll use GPOs?	We designed our AD to meet our business goals. We didn't really take GPOs into consideration.
What GPO functionality do you want to use in your environment?	We know we want to use desktop lockdown (Administrative Templates) and Folder Redirection, but we're not sure about other features. We want to be able to seamlessly migrate our NT 4.0 system policy lockdown to Administrative Templates.
Will you create and manage AD objects (and thus GPOs) from a central group within your organization, or will management be delegated across many business units?	We only have one central IT group for support, so it'll handle all AD and GPO administration centrally.
Will you create GPOs and link them right to the container where they apply (say, at an OU)? Or will you link them at a higher level (such as at the domain level) and use security-group filtering to control their effect?	We want to avoid too much complexity, but we need total control of GPO management.
How will you manage aspects of GPO administration such as change control, RSoP, and GPO processing performance, and will the GPO delegation design you've created allow such management?	We haven't thought about any of these aspects, but we want to avoid them if possible.

Table 4.1: Company A's answers to the five GPO design questions.

Given the answers shown in Table 4.1, what is the best GPO design for Company A? Of course, there are probably a couple of answers to this question, but let's start by looking at its AD design. Figure 4.1 shows what Company A's AD infrastructure will look like.

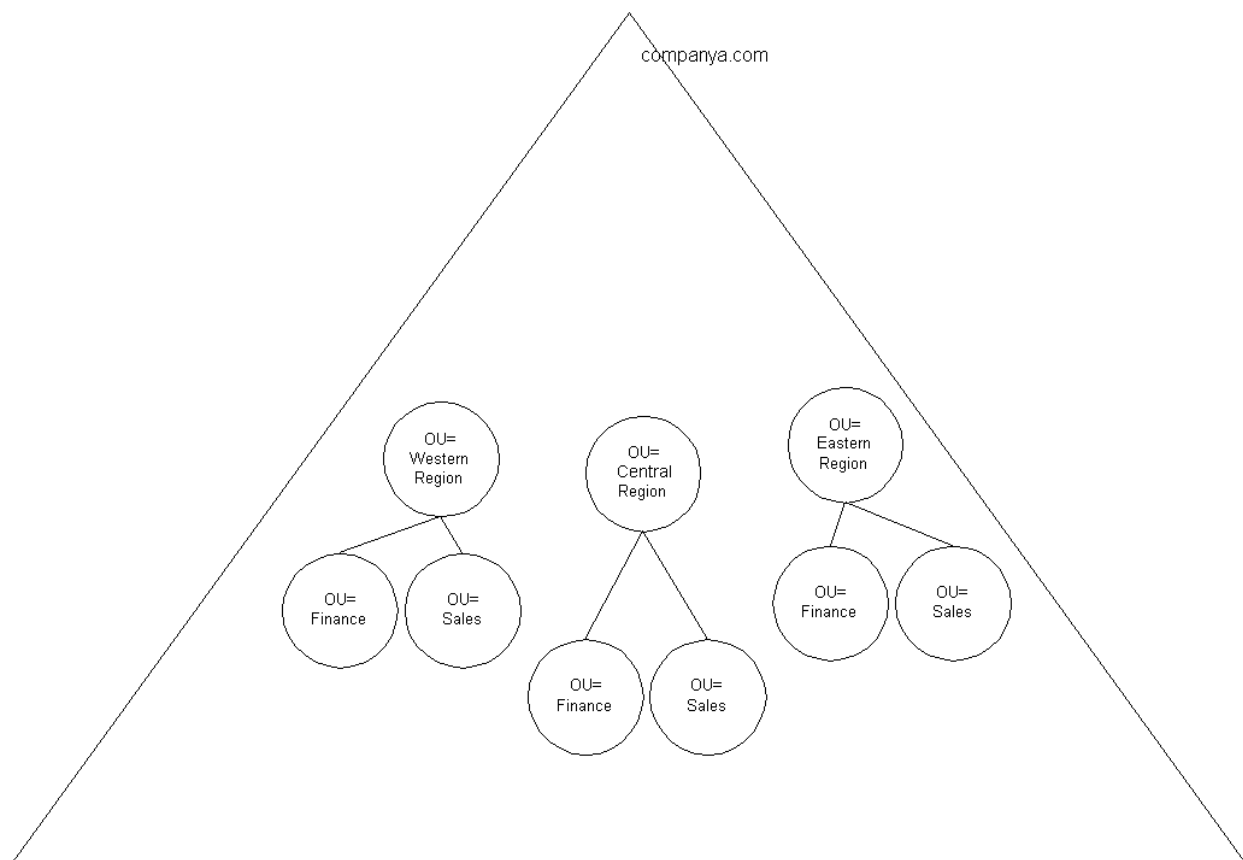


Figure 4.1: Company A's AD namespace.

As you can see in this figure, Company A's AD domain is called **companya.com**. It contains three top-level OUs: **WesternRegion**, **CentralRegion**, and **EasternRegion**, and within each of these top-level OUs are two OUs called **Sales** and **Finance**. User and machine accounts reside in each **Sales** and **Finance** OU under each region. Each regional OU structure is physically located in a different part of the country, and thus each region is in a different AD site from the other.

Company A wants to take advantage of the Administrative Templates and Folder Redirection features of GPOs. In fact, it relied heavily on system policy in its NT 4.0 environment to lock down the desktops of its users based on their membership in a user group. Company A differentiated desktop lockdown in system policy among the following six user groups:

- WesternSales
- WesternFinance
- CentralSales
- CentralFinance
- EasternSales
- EasternFinance

The `ntconfig.pol` file used in Company A's NT 4.0 environment is shown in Figure 4.2 below.



Figure 4.2: Company A's NT 4.0 system policy file.

In NT 4.0 system policy, a single policy file could apply different desktop-lockdown settings to different user groups. Unfortunately, in Win2K Group Policy, each GPO only holds a single set of Administrative Template settings. Company A can filter the effects of a particular GPO to only apply to a particular user group, but to gain the same lockdown functionality that its `ntconfig.pol` file provided, the company may need to build a single GPO for each user group that it wants to apply policy to. This means six GPOs in Win2K compared to one NT system policy file in NT 4.0. This may sound less than ideal, but there are steps the company can take to mitigate the disadvantages of this approach, and I'll present them shortly.

Now let's look at what Company A needs to implement Folder Redirection. Remember that there are two types of Folder Redirection in Group Policy—basic and advanced. In basic redirection, all users who are subject to the GPO containing the policy have the prescribed folder redirected to a single destination (for example, `\\server\share\%username%`). In advanced redirection, you can redirect the folder to different destinations (such as different servers or shares) based on group membership.

Company A would like to redirect the My Documents folder to servers local to each user group. Given that requirement, it can actually choose one of three GPO designs. It can create:

- A single GPO, linked to the `company.com` domain, that uses advanced redirection based on user groups defined in each OU where users are to be redirected (for example, `WesternRegion\Sales`, `EasternRegion\Finance`, and so on).
- Three GPOs linked to each regional OU (Western, Central, and Eastern), then use basic redirection to redirect users within each regional OU structure.
- Six GPOs, each of which is linked to each of the regional Sales and Finance OUs—again using basic redirection. This design gives administrators the ability to redirect each OU to a different location without using advanced redirection.

In the new AD structure, instead of having user groups to differentiate users, Company A can segregate its users into the various OUs. This is a very common trade-off as you're designing and deploying both AD and GPOs. It also brings up a key point—namely, groups and OUs perform complementary and overlapping functions. You can segregate users (and computers) by placing them in OUs or by placing them in groups (in which case they may span multiple OUs).

OUs are a more logical place to segregate resources because they don't require explicit maintenance. That is, when you create a new user in an OU, all of the GPOs linked to that OU (and inherited by the OU) are automatically applied. However, if you use security groups to segregate users or computers and filter the effects of GPOs, your administrators need to remember, each time a new user is added, to place the user in the correct groups. In either case, this trade-off will affect how complex your GPO design needs to be and how complex your AD delegation model is.

✍ There is an inherent trade-off in AD between using OUs to segregate users and computers and using security groups. In some cases, one or the other may better meet your needs, but for the sake of simplicity, always organize users and computers in OUs rather than security groups.

Two Implementation Options

With this background information, let's look at a couple of possible GPO implementations that will meet Company A's needs. Figure 4.3 shows the first option.

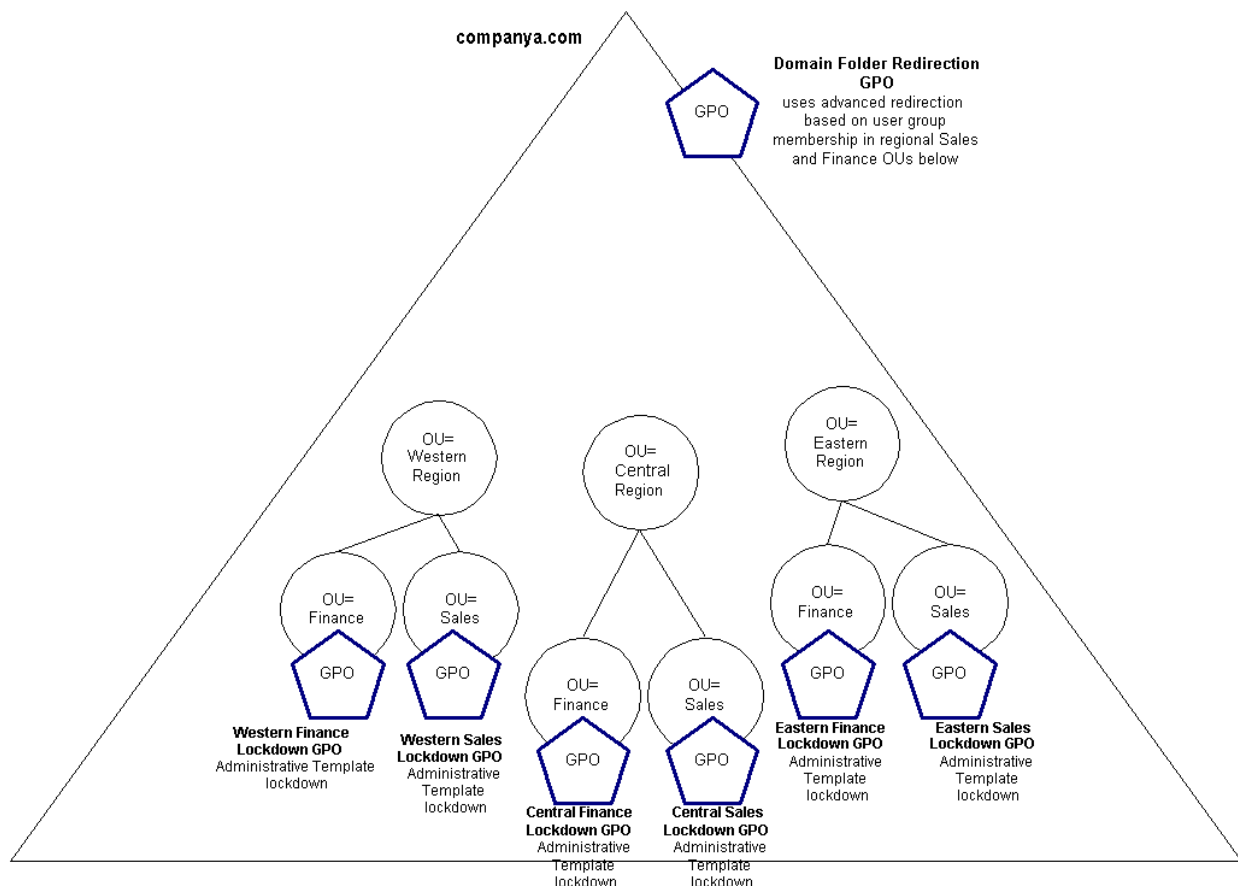


Figure 4.3: A possible GPO implementation for Company A.

Figure 4.3 shows a total of seven GPOs (shown in dark blue) defined in Company A's AD domain. The first GPO, called Domain Folder Redirection GPO, is linked to the domain level and uses advanced redirection based on user groups defined in each of the regional Sales and Finance OUs below it. Of course, I could have linked this GPO lower in the AD hierarchy—at the OU level, for example. To do that, however, I'd have needed three GPOs (one each for the Western, Central, and Eastern regional OUs) to accomplish the same task as one, and I'd still have needed to use advanced redirection to redirect each Sales and Finance OU to a different location.

Now let's look at the six other GPOs in Figure 4.3. Because my requirements were to emulate the desktop-lockdown functionality in Company A's NT 4.0 environment, I had no choice but to create a separate GPO for each OU where the user accounts reside. If I'd placed three GPOs at the regional OU levels, I wouldn't have been able to differentiate Administrative Template policy for the two OUs (Sales and Finance) in this regional OU. Thus, I'm forced to create six nearly identical GPOs and link them to each OU where I want to set Administrative Template policy. Fortunately, each user or computer in the Sales and Finance OUs only has to process, at most, two GPOs—the domain-linked Folder Redirection GPO and the OU-linked Administrative Template GPOs.

As with many things, there is no one correct answer. Figure 4.4 shows another option that will meet Company A's needs.

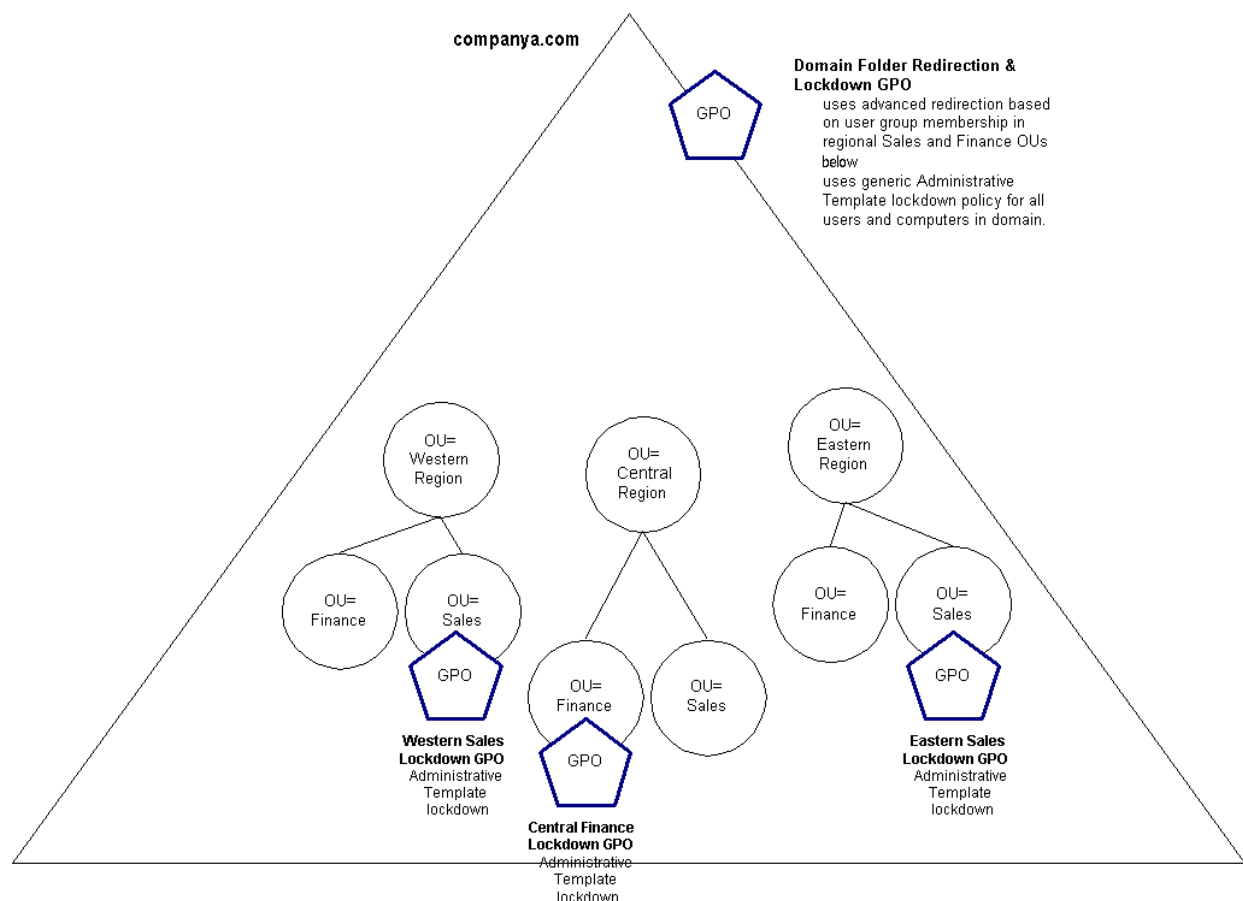


Figure 4.4: A second possible GPO deployment for Company A.

Let's suppose that Company A's administrators looked at the GPO design in Figure 4.3 and thought that there were too many GPOs to administer. In that figure, each GPO only did one thing—either Folder Redirection or Administrative Template lockdown. With only a slight change in requirements, Figure 4.4 reduces the number of GPOs required and presents a more reasonable management picture. What has changed?

In Figure 4.4, the Domain Folder Redirection GPO gained some functionality—namely, Administrative Template lockdown. On taking a closer look at the lockdown that Company A required, I saw that 90 percent of the policies were identical among all of the OUs subject to that policy. So I applied that 90 percent to the domain-linked GPO—thus applying that lockdown to all users. For those users and computers that required a slightly different lockdown, I created the requisite GPOs and linked them to the OUs in question. Only Western Sales, Central Finance, and Eastern Sales required a different lockdown from the domain-based policy.

Of course, if each OU really requires some differentiated lockdown, Figure 4.3 is probably my better option. The point in Figure 4.4 is that by making some compromises and combining multiple functionality in a single GPO, I was able to reduce the number of GPOs that I needed to manage by three.

Both of the solutions that I've presented for Company A are possible because it's centrally managed and has total control over creating and managing its GPOs. Life gets more complicated in Scenario B below, as I present a company with more complexity and a more distributed administrative environment.

Scenario B—Multiple AD Domains

In this scenario, I've created a large multinational corporation called ABC Corp., which wants to implement a multi-domain AD forest. Table 4.2 shows the responses for this scenario to the five questions listed in "Integrating the Design into Your AD Infrastructure" earlier in this chapter.

Question	Answer
Does your AD design take into consideration how you'll use GPOs?	GPO usage was a key part of how the AD namespace was designed.
What GPO functionality do you want to use in your environment?	Most GPO functionality will be required, especially Folder Redirection, Administrative Template policy, IE Maintenance, and Software Installation.
Will you create and manage AD objects (and thus GPOs) from a central group within your organization, or will management be delegated across many business units?	Creating GPOs will be handled centrally, as will linking GPOs to domain and sites. Special GPOs, such as those for domain-wide security, will also be managed centrally. However, the task of linking GPOs to particular OUs will be distributed to regional AD administrators. Administrative Template-based policy will be used to restrict which administrators can load which GPO snap-in extensions.
Will you create GPOs and link them right to the container where they apply (say, at an OU)? Or will you link them at a higher level (such as at the domain level) and use security-group filtering to control their effect?	GPOs will be linked to the container closest to the users or computers where they're applied. GPO security-group filtering will be used, but only sparingly.

Question	Answer
How will you manage aspects of GPO administration such as change control, RSoP, and GPO processing performance, and will the GPO delegation design you've created allow such management?	All GPOs will be version-controlled using their display name. RSoP tools will be used to test policy changes and determine the impact on users or computers before deployment. To maximize processing performance of GPOs, no user or computer will be subjected to more than ten GPOs.

Table 4.2: ABC Corp.'s answers to the five GPO design questions.

Table 4.2 gives us a lot to talk about. Let's start by looking at the AD namespace for ABC Corp., as shown in Figure 4.5.

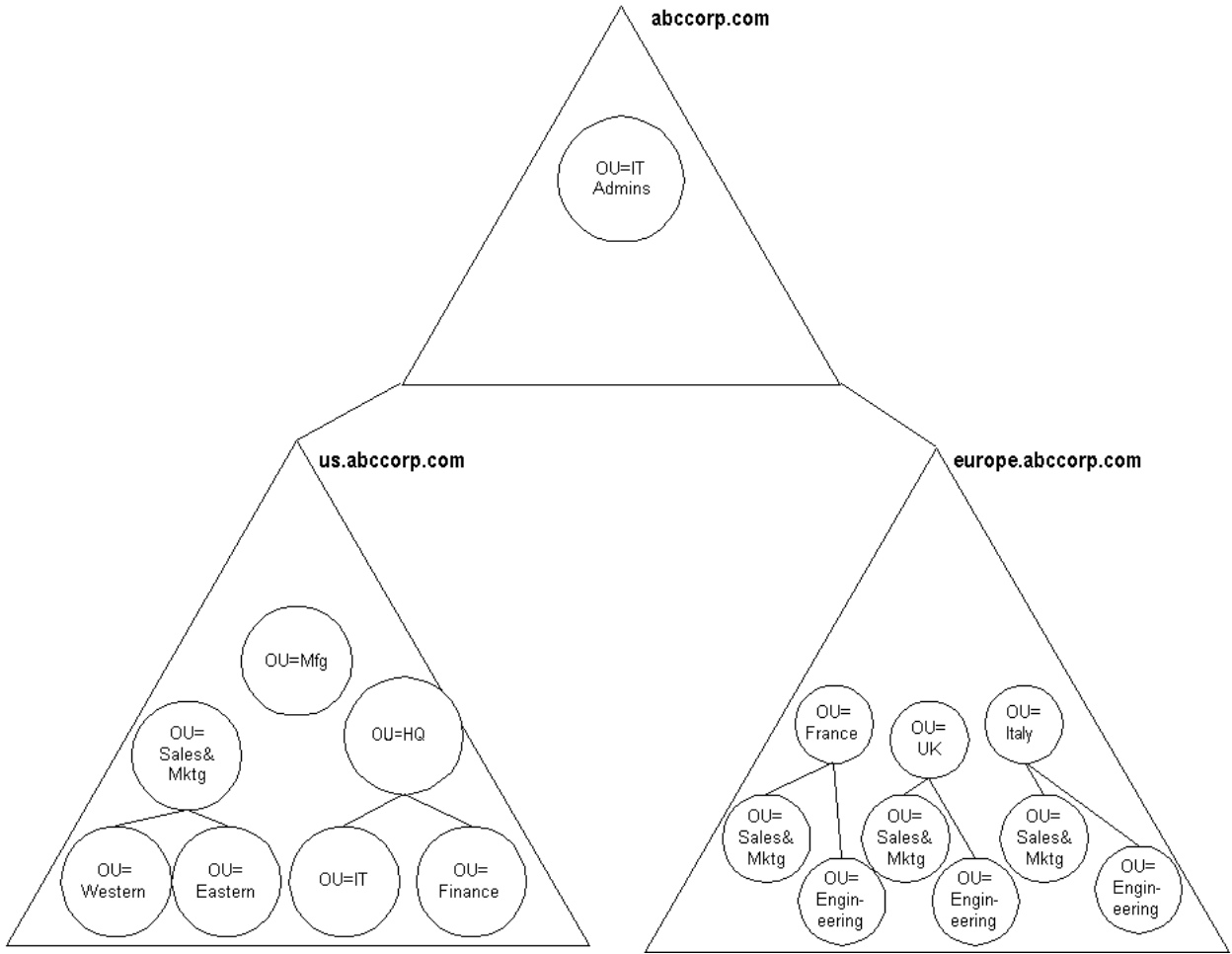


Figure 4.5: ABC Corp.'s AD namespace.

As you can see, there's a lot going on in ABC Corp.'s AD infrastructure. First, the forest contains three domains—abccorp.com is the forest root domain, and us.abccorp.com and europe.abccorp.com are children of the root domain. In the forest root domain, one OU is defined, called ITAdmins, where the forest administrators' user accounts are stored. Each of the two child domains contains a number of OUs of differing structure. In the US child domain,

there are three top-level OUs called Mfg, Sales&Mktg, and HQ. Under Sales&Mktg are two OUs called Western and Eastern because the Sales and Marketing division is divided into two regions. Under the HQ OU are two sub-OUs called IT and Finance. They hold users and computers in the IT and Finance departments at ABC Corp.

Now, if we shift over to the Europe domain, we see a different OU organization. The three top-level OUs are organized by country—France, UK, and Italy. Under each of these top-level OUs are two OUs—Sales&Mktg and Engineering. As you can see, this AD forest appears to have no consistent approach to how OUs are built and organized. In the US domain, top-level OUs are a combination of department and location, whereas in Europe, the three top-level domains define countries, and departments are housed in the sub-OUs under each country.

This seeming mix of organization has a rationale. The OUs are organized according to the various IT support organizations in the company. For example, in the US domain, the Sales&Mktg OU is administered by a separate group of administrators from the one that administers the HQ OU. By organizing OUs according to their administrative authority, it's easier to delegate administration of AD (and GPO) management cleanly along OU lines. In Europe, one support organization is responsible for each of the top-level OU hierarchies. As you'll see later in this section, this lends the GPO design a degree of simplicity that's required for these users and computers.

Now you're probably thinking, "I thought this was a chapter on GPO design, not AD design." My goal in describing this AD design in detail was to show that the AD and GPO designs are linked and one cannot be discussed without the other. Now I'll discuss GPOs for ABC Corp.

As Table 4.2 pointed out, ABC Corp. wants to use several features of GPOs. It also wants GPOs to be created centrally but wants to be able to delegate the linking of some GPOs to OU administrators. Given the complexity of its AD namespace, this company wants to minimize the use of security filtering of GPOs by linking them close to the users and computers that they're intended for.

By "close," I mean that if the company creates a GPO to lock down the desktop of the HQ\Finance users in the US domain, it wants to link that GPO to the Finance OU under HQ. Otherwise, if the company links the GPO higher in the AD tree—perhaps at the domain level—it has to use security groups to ensure that the lockdown policy defined in that GPO is only processed by an HQ\Finance user group. This is another example of the trade-off between using OUs or using security groups to segregate users—in this case, for the purpose of filtering Group Policy.

Two Implementation Options

Figures 4.6a and 4.6b show two GPO design options for ABC Corp.

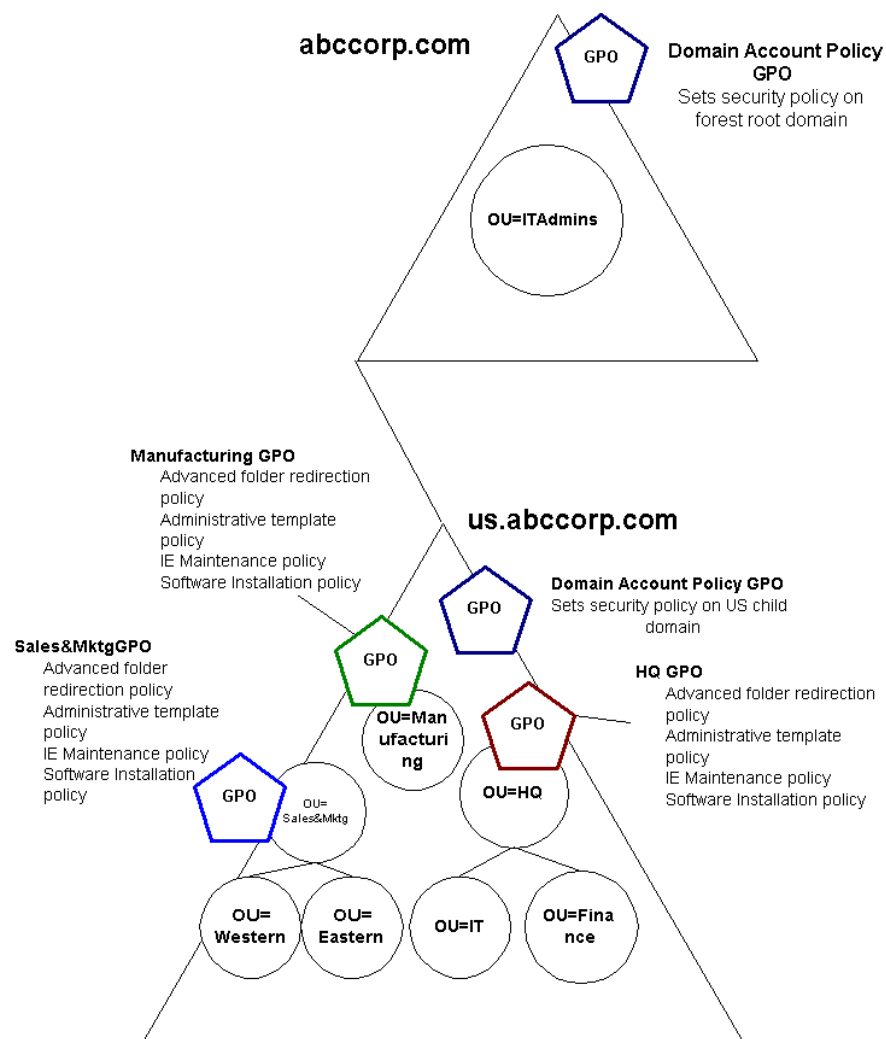


Figure 4.6a: In the first design option, the US subdomain is part of the GPO design.

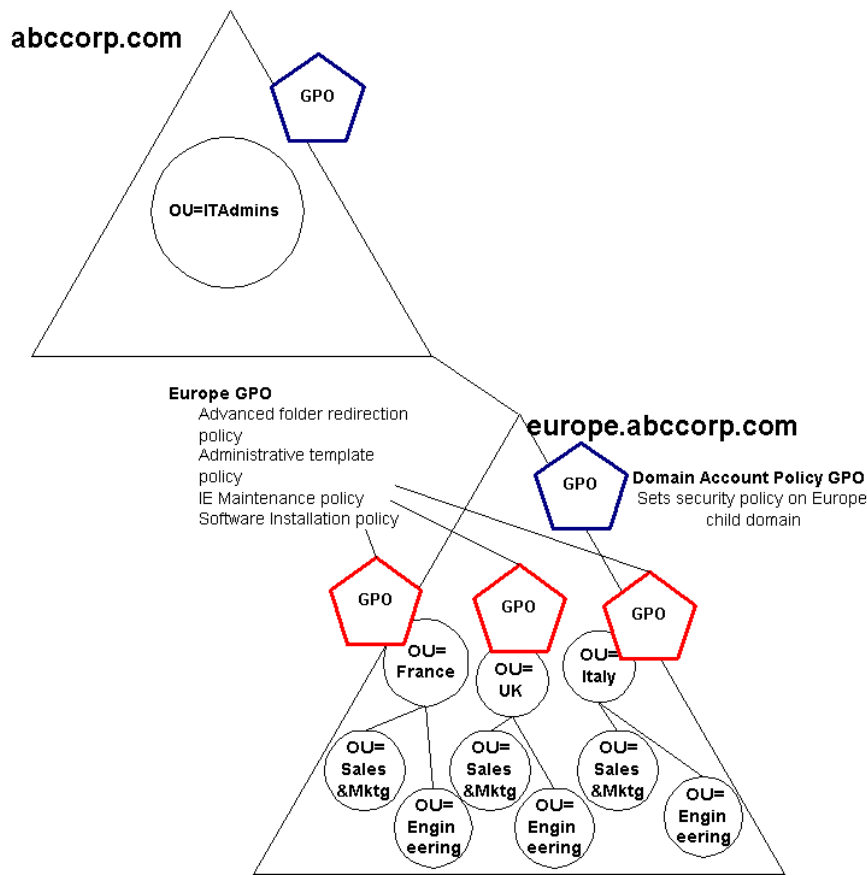


Figure 4.6b: In the second design option, the Europe subdomain is part of the GPO design.

There is a lot going on in the figures above, so let's walk through them. The colors of each set of GPOs represent the administrative group that is responsible for managing them. All GPOs were initially created by the forest administrators (represented by dark blue). These forest administrators also manage the domain security policy linked on each domain in the forest. Each domain has its own security policy. I could have easily created a single security GPO in the forest root domain (abccorp.com) and linked to it from each of the two child domains, but because abccorp.com has different security requirements from the US and Europe, I would have had to create three GPOs.

In the US domain (Figure 4.6a), three GPOs (light blue, green, and dark red) are linked to each of the three top-level domains. This indicates that a different IT administrative group maintains each GPO. However, the same set of GPO functionality is defined in each GPO—namely, Folder Redirection, Administrative Template policy, Software Installation, and IE Maintenance. There may be subtle differences among the GPOs, but the important thing to remember is that a different group manages each one.

Of course, the approach I've shown in Figure 4.6a assumes that each sub-OU under the top-level OUs needs the same policy settings as its neighbor. This may not be the case, and if so, the administrators of this domain may need to create additional GPOs, linked to either the top-level OUs (with security-group filtering to control who gets what policy) or the lower-level OUs (where the users and computers physically reside).

In the Europe domain (Figure 4.6b), the GPOs are red and have the same name. They're the same color because only one IT group administers this domain. In addition, because the GPO needs of each top-level OU and its subordinate sub-OUs are identical, only one GPO is actually defined (hence the same name for each), and it has then been linked to by each of the three top-level OUs. Of course, this is generally the exception rather than the rule. It's very likely that the needs of any country will differ at least slightly from those of the other two. However, the concept of linking to a single GPO from multiple OUs is a common one, and I wanted to represent it here.

Handling Delegation Requirements

As was the case with Company A, there are probably a few variations to Figures 4.6a and 4.6b that would meet the needs of ABC Corp. But from a GPO delegation perspective, ABC Corp. has different requirements than Company A. As I mentioned in Table 4.2 above, the forest administrators wanted to be able to create all of the GPOs in the forest but leave the task of linking and managing OU-based GPOs to regional or departmental administrators. (Remember from Chapter 3 that you can create a GPO without linking it to a container.)

It's important to know how your AD design will lend itself to these delegation requirements. (I'll discuss the challenges of GPO delegation in greater detail in Chapter 5.) In the meantime, keep in mind that the ability to link and un-link a GPO from an OU or domain (or site) is granted by a right on the particular container (either site, OU, or domain), while the ability to edit that GPO is granted by modifying permissions on the GPO (actually, on the Group Policy Container, or GPC) itself using the Security page of that GPO (see Figure 4.7).

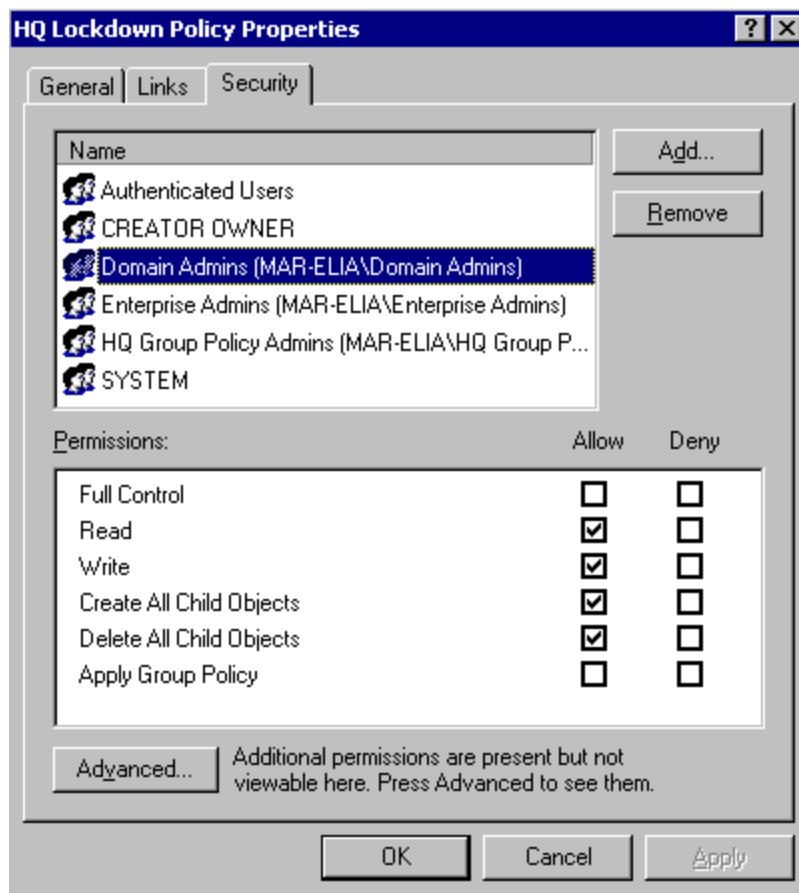


Figure 4.7: The security properties that control delegation of administration on a GPO.

To see the Security page, right click the GPO name in the Group Policy MMC snap-in, then choose Properties from the context menu. In the Properties dialog box, click the Security tab. (I described this dialog box in “Using Security Groups to Filter the Processing of GPOs” in Chapter 3.) To control who processes a policy, you simply need to grant the Read and Apply Group Policy rights to that user, machine, or group. To control who can edit a policy, you need to grant other rights to that policy (such as Read, Write, Create All Child Objects, and Delete All Child Objects). In the case of ABC Corp., if I’m interested in granting editing rights to the HQ GPO for HQ administrators, I’d modify the security permissions on that GPO to grant a group that includes the HQ admins the rights to modify that GPO. I’ll talk more about GPO delegation later in this chapter (see “Managing Change Control”).

Multi-domain Considerations

Scenario B is different from Scenario A in one important way—it includes multiple AD domains. From a GPO perspective, multiple domains are important because they give administrators the opportunity to link to a GPO that is stored in a domain other than their own. This is, of course, perfectly legitimate. As I mentioned above, ABC Corp. could have created a single GPO in the forest root domain to set security policy, then linked each of the two child domains to that GPO.

However, there are a couple of reasons why you should seek to minimize the amount of cross-domain linking of GPOs that you use.

- Whenever a user or computer must traverse domain boundaries to resolve the GPC and Group Policy Template (GPT) of a remote GPO, a certain amount of extra overhead occurs as that user or computer traverses the associated trust relationships, passing its credentials from one domain to the next. This slows down the processing of a GPO that is remotely linked.
- Linking across domain boundaries introduces complexity into your GPO design and makes it harder to troubleshoot problems should they arise. Because the domain is remote, the administrator linking to the GPO may not know anything about its management—for example, whether it's currently functional, in the process of being changed, and so on.

Site-Linked GPOs

I haven't spent any time yet discussing linking GPOs to AD sites. As you know, *sites* are AD objects that help control directory replication among domain controllers. You, as the administrator, define AD sites and the Internet Protocol (IP) subnets that belong to each site. Then you define site links to connect sites together to form an AD replication topology. You can link GPOs to sites just as you link them to domains or OUs. When a GPO is linked to a site, all users and computers that reside in that site are subject to that GPO. You link GPOs to sites using the AD Sites and Services MMC snap-in (see Figure 4.8).

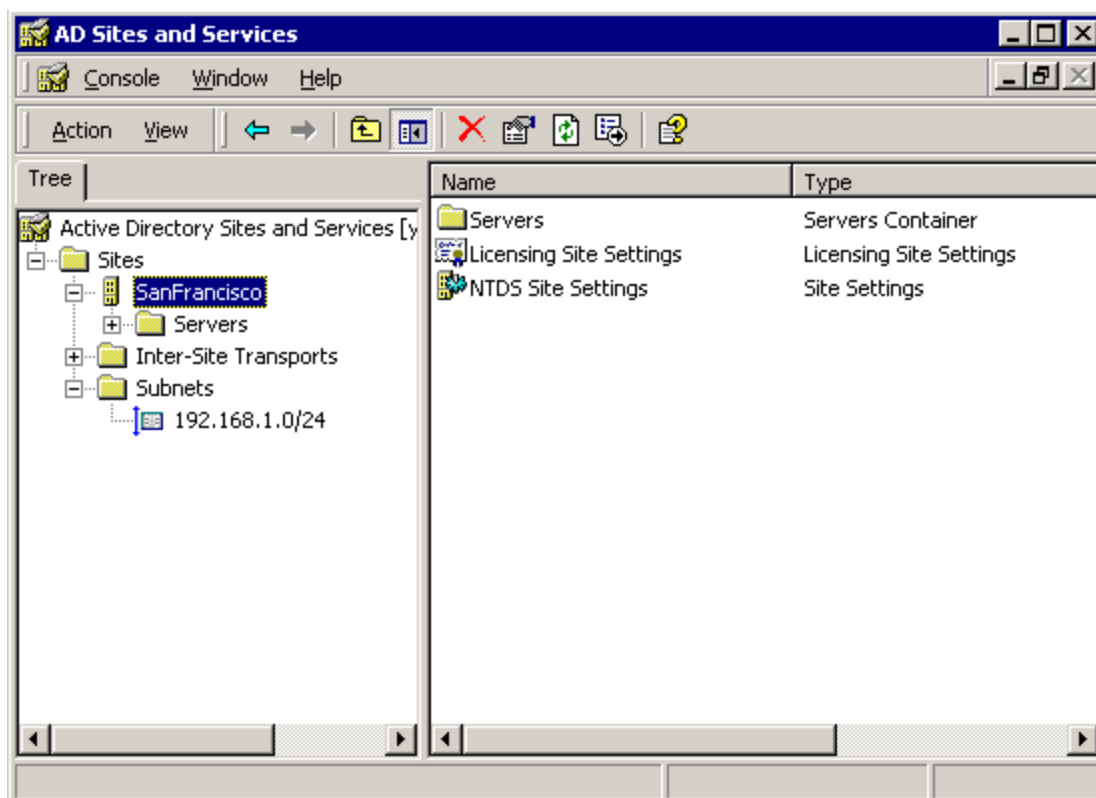


Figure 4.8: Using the AD Sites and Services MMC snap-in to link GPOs to sites.

In this figure, the SanFrancisco site contains the subnet object 192.168.1.0. If you right-click this site object, then choose Properties from the context menu, the Properties dialog box appears. When you click the Group Policy tab, you can create a new GPO that is linked to this site or add a link to an existing GPO defined elsewhere.

The important question for this chapter is: Why would anyone use site-linked GPOs? Indeed, I've found very few circumstances where GPOs linked to a site are actually valuable. Remember that sites can span multiple AD domains, so you can have devices from multiple domains in a particular site. However, GPOs themselves are stored in a single domain. This means that if you define a site-based GPO that is stored in abccorp.com, for example, it can be processed by a computer or user residing in a different domain in the same site. As I discussed in "Multi-domain Considerations" above, whenever GPO processing has to cross domain boundaries, you're likely to get some degradation of performance.

Aside from that, however, you may find it difficult, as I have, to justify why a GPO would need to be linked to a site as opposed to a domain or OU. Because sites are really just collections of subnets, you can think of site-linked GPOs as a way of providing Group Policy based on where on your network a particular machine resides. One valuable use of such policy might be the machine-specific IP Security (IPSec) policy I described in Chapter 1. You might want to set policy to specify that any machine that is on your corporate network, on a particular set of IP subnets, must use IPSec to communicate with other computers. Similarly, you might have a number of mobile users who always dial into your network on a well-known IP subnet, and you might want to provide them with some specific mobile policy that applies only to them when they're dialing in.

Unfortunately, there are a couple of challenges to site-linked GPOs. First, because in the Site, Domain, Organizational Unit (SDOU) order of precedence, site-linked GPOs are processed first, they can be easily overwritten by GPOs linked to domains or OUs, which are processed later. This means that for any policy you set in a site-linked GPO to stand, it needs to be unique (that is, only defined there) for all machines and users processing it.

Second, because it's more likely that machines will change their IP subnet (for example, when they move or are mobile), you need to ensure that as machines move from site to site, the GPOs that are applied to them provide a consistent and expected experience. For example, in certain sites, a GPO linked there might lock down aspects of your users' desktops that they're not used to. You almost need to think in advance about how a machine or user will be affected as it moves around before you can define effective site-linked policy.

Given these caveats, however, site-linked policy can be a powerful tool in situations where proximity on your corporate network is a significant enough factor to determine which policy applies.

Adopting GPO Naming Conventions

Another aspect to consider as you're designing your GPO implementation is the naming convention for your GPOs. As you saw in Chapter 2, GPOs are uniquely identified in AD by their Globally Unique ID (GUID). However, every GPO has a *display name*, or friendly name, that you see when you're opening one for viewing or editing with a tool like the AD Users and Computers MMC snap-in. It's this name that we're interested in when we talk about GPO naming.

Just as with other objects in your AD infrastructure, it's important to come up with a good, standardized naming convention for GPO display names. The trick is that the convention must take into consideration that, while the display name is stored with the GPO itself, GPOs can be linked to any number of containers in your AD. For example, if you name your GPOs using a convention that includes the name of the container to which they're originally linked (such as Finance OU Lockdown Policy), then link that GPO to other containers, your convention may fall apart.

For that reason, I recommend coming up with a "container-neutral" naming convention. This means that in Scenario B above, for example, instead of naming my GPOs after the containers they were linked to, I should have given them names that are less container-specific. For example, instead of having an HQ GPO, a better name would be something like FR,SI,AT,IE #1 GPO. This code, if you will, simply lists the functions in the GPO that I'm using (FR=Folder Redirection, SI=Software Installation, AT=Administrative Templates, and so on). The #1 indicates that there may be more of this kind of GPO, such as #2, #3, and so on.

The key here is to keep the names as generic, but as descriptive, as possible. Of course, if you don't plan to link GPOs from all over your AD infrastructure, it's perfectly reasonable to give a GPO a location-specific name (such as Europe Lockdown GPO). Remember that the display name for a GPO can be changed to your heart's content because it's the GPO's GUID that really matters to AD.

Another tip you might want to consider is including some kind of version information in a GPO's name. As you know from Chapter 2, GPCs and GPTs keep their own version-control information to ensure that they're synchronized across your AD infrastructure. To maintain change control over GPOs, you can include version numbers in your naming convention to ensure that administrators who are linking to a particular policy know that they're linking to the correct version. Figure 4.9 shows GPOs that use version numbers in their names.

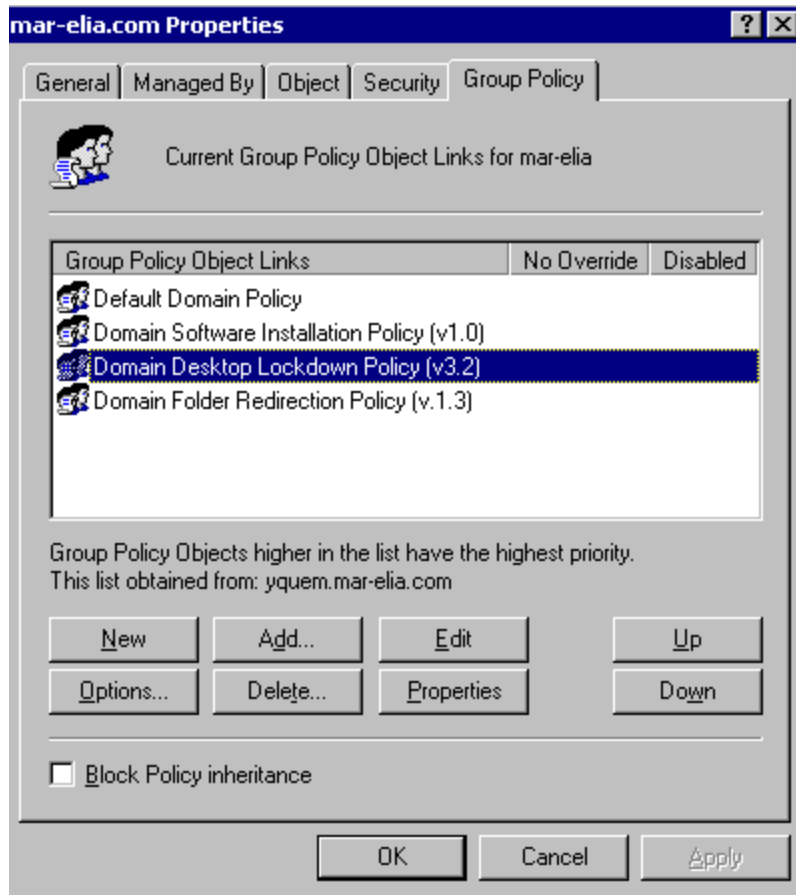


Figure 4.9: Using version numbers in GPO display names.

As you can see in this figure, the names of the three GPOs contain version information, which tells administrators what they're dealing with. Each time a GPO is edited, the person who makes the change needs to change the GPO's display name to increment its version number. In addition, the names of the GPOs themselves describe the functions that they perform (for example, Admin Template, Software Installation, and Audit policy). Of course, you can always use the revision information displayed on the General page on the GPO (see Figure 4.10) to keep version-control information.

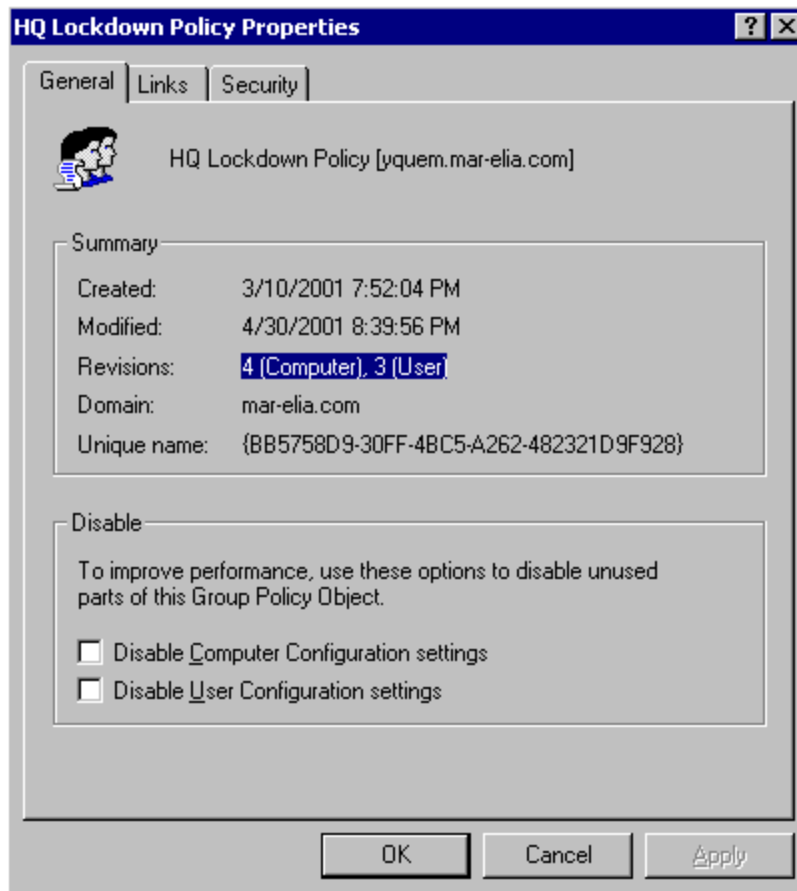


Figure 4.10: *Displaying revision information in a GPO.*

The bottom line is that you need to devise a naming convention that you apply consistently to all GPOs and maintain as they change.

Maintaining Change Control

And speaking of change—change control is another issue that you’ll have to face in a GPO implementation, especially in large AD infrastructures. Unlike normal software-release or configuration-management processes, there are no built-in mechanisms for decent change control of GPOs. You need a way of knowing who did what to which GPO and when they did it.

You can use AD and file-system auditing on SYSVOL to determine at least part of this information, but this entails wading through security logs on potentially multiple domain controllers to figure out what happened. Also, as I mentioned earlier in this book, the GPO editing tools don’t provide any staging of GPO changes; you can only test your changes in a non-production AD infrastructure. Once you click Apply or OK to a change, it’s committed and starts to replicate to all domain controllers.

In Chapter 3, I talked about a technique for creating GPOs without linking them to a container. You can use this no-link approach to phase new GPOs into production. This is a good approach for ensuring a phased implementation of new GPOs, but it isn’t useful for existing GPOs. When

you have to change an existing GPO, you have few choices to ensure that it doesn't adversely affect your production users.

The first and most obvious step to take is to have a test AD forest available in your environment. This test forest would mimic some or all of your production forest and would provide you with an environment for trying out the effects of new GPO changes. Once a GPO change is validated in the test forest, you can apply it to your production environment with relative safety.

Another technique you can use when making multiple changes to GPOs in a large AD infrastructure is to first disable a GPO that you plan to change. Remember that in Chapter 3, I described how you can disable a GPO to prevent it from being processed. Figure 4.11 shows the dialog box under GPO Options where you can disable a GPO.

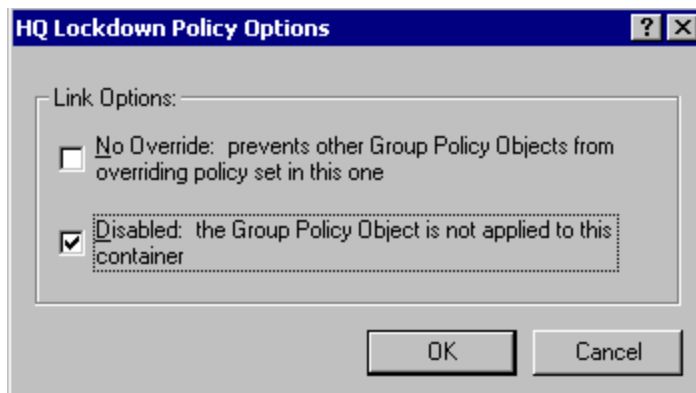


Figure 4.11: Disabling a GPO while editing.

If you're making multiple changes to a GPO, you don't want users to be processing it—and getting unexpected results. Disabling the GPO before editing ensures that no user is processing it while you're making changes.

✎ Disabling a GPO doesn't remove the policy that was previously applied to it. It simply tells the computer or user subject to it that it's not available during the current processing cycle. Once you complete your changes, enable the GPO using the dialog box in Figure 4.11, and during the next processing cycle, your users and computers will receive the updated policy.

GPO change-control issues become more challenging if you've delegated administration of GPOs across your infrastructure. You need to make sure that all of your administrators are aware of the implications of editing GPOs. This is especially true when GPOs are linked to from multiple containers.

My recommendation is to only delegate administration of GPOs that aren't linked to from multiple containers. For example, if an OU administrator wants to maintain his or her own GPO, make sure that that GPO is only used in that OU structure and isn't linked to by other OU administrators. If you need to use GPOs that are linked to from multiple containers, ensure change-control consistency by keeping them controlled by a centrally managed group. Despite AD's strong handling of delegated administration, GPOs are one of those aspects of the technology that may better be delegated slowly and in a limited way.

Tying It All Together

I've presented two GPO design scenarios, and I've raised issues about the related challenges of designing and managing a GPO implementation. Now I want to tie all this together in a few general guidelines for creating a good GPO design. The following list summarizes the points I've discussed in this section:

1. Minimize the complexity of your initial GPO design until you're comfortable with the technology.
2. When you need to manage one policy separately from the others, consider creating "single-function" GPOs (for example, just security or just Administrative Templates).
3. If you're working in a multi-domain AD environment, for performance and troubleshooting reasons, minimize cross-domain linking of GPOs.
4. Minimize the number of GPOs that need to be processed by a user or computer. Use some of the techniques described earlier in this book—such as disabling computer- or user-configuration policy for GPOs that don't use it to speed up the processing of existing GPOs. Less than ten GPOs is a nice round number to aim for. If you need more GPOs, make sure that only those GPOs that absolutely have to be processed by a user or computer are indeed processed, limiting others using techniques like disabling user or computer configuration or security-group filtering.
5. Always try to link GPOs as close to their intended target audience as possible. For example, if you need to set security policies for all machines in a domain, link the GPO at the domain level. But if you're setting security policy on a set of machines that resides in a single OU, link at that OU only. (The alternative to this approach is to link higher in the AD hierarchy, then use security groups to filter the effect of a GPO. However, this is less desirable because it makes troubleshooting effective policy more challenging and less obvious.) Finally, when you need to enforce network-specific policy such as IPSec, link GPOs to sites.
6. Delegate administration of GPOs only to the extent that is needed to support your business goals. In general, it's a good idea to keep as many of your GPOs centrally managed as possible to ensure good change control and consistency of use. And because GPOs (that is, their component GPCs and GPTs) are replicated to all DCs in a domain, it's important that forest-wide administrators know what's being done to their infrastructure.
7. Develop a good and consistent process for naming and changing GPOs, then ensure that everyone in the forest follows it.

Considering Technical Issues

In addition to creating a good initial design of your GPO infrastructure, you'll face some technical issues as you deploy GPOs. These issues may significantly affect your design or its performance, so it's important to keep them in mind.

Asynchronous, Slow Network, and Periodic Background GPO Processing

Some types of processing in GPOs can affect your design process. These effects are summarized in Table 4.3 below.

This Feature	Produces This Effect
Asynchronous GPO processing	GPOs are normally processed synchronously. That is, a computer processes all GPOs before it presents a user dialog box, and a user processes all GPOs before the Desktop is presented. Using asynchronous processing speeds the machine startup and user logon process, but it doesn't guarantee that the necessary GPOs will have finished processing by the time each event occurs.
Slow-network detection processing	When a machine starts up and a slow network link is detected, certain GPO policy is disabled (for example, Folder Redirection and Software Installation). This limits the effectiveness of these policies for roaming users.
Periodic background processing	Certain policy (such as machine security and Administrative Templates) isn't just applied when a machine starts up or a user logs on. It's re-applied periodically thereafter (by default, every 90 minutes).

Table 4.3: GPO processing that can affect your GPO design.

As you plan your GPO rollout, it's important for you to recognize the effects of these three types of processing. For example, if many of your users access your AD infrastructure over slow links—by default, less than 500 kilobits per second (Kbps)—some policies, like Software Installation and Folder Redirection, won't be applied. Using these policies may therefore not be effective in your environment. Or if you're worried about the processing performance of GPOs in your environment, you can enable asynchronous GPO processing for the machine or user (or both), thus reducing the wait time the machine incurs processing GPOs.

Finally, it's important to know that background processing occurs for certain policy, such as security and Administrative Templates. This means that even if your users attempt to foil a particular policy on their local machines, their “futz” will be undone later when policy is re-applied.

Controlling Their Effects

You can control the effects of these three types of processing using Administrative Template policy in a GPO. These GPO processing policies are found in the Computer and User sections of Administrative Templates, under Administrative Templates\System\Group Policy. Figure 4.12 shows an example of the machine-specific GPO processing policies available.

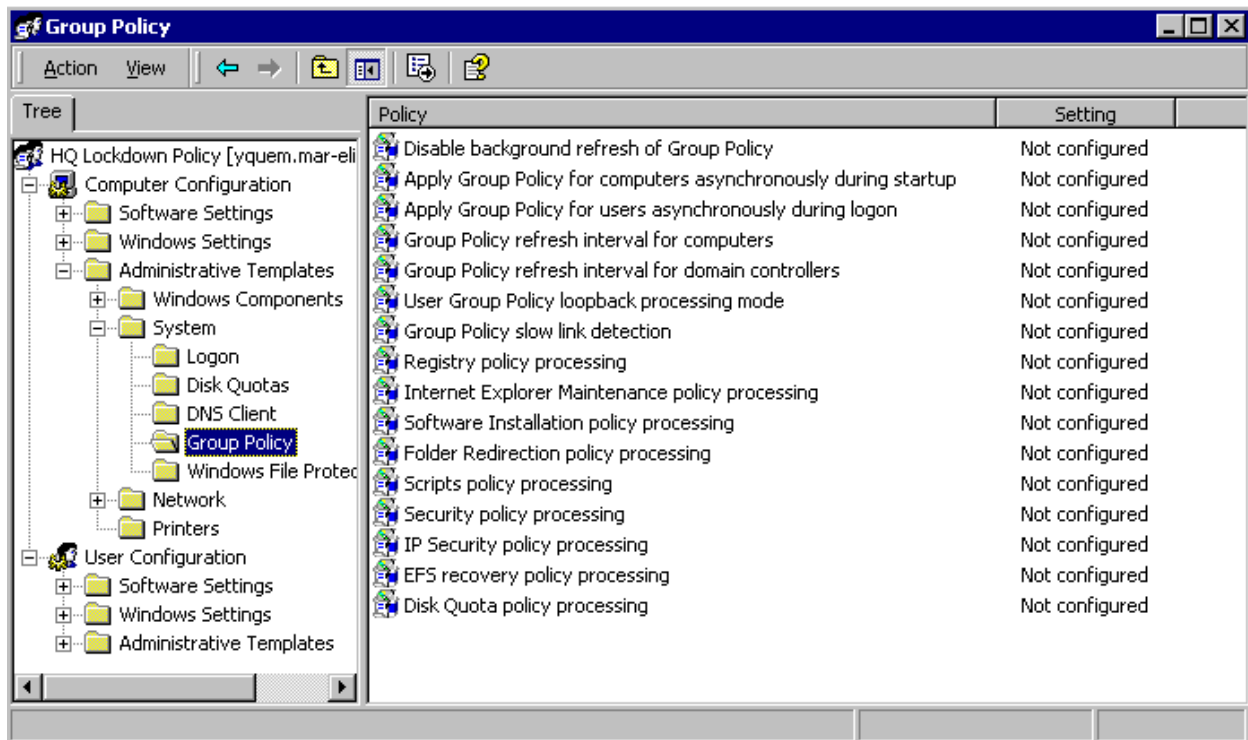


Figure 4.12: The machine-specific GPO processing policies available in Administrative Templates.

If you view each of the policy items available in this policy section, you can change the default behavior for a number of GPO processing options. For example, if you don't want Folder Redirection to be disabled when a slow network link is detected, select Folder Redirection Policy Processing. In the Properties dialog box, click Enabled, then click Allow Processing across a Slow Network Connection (see Figure 4.13).

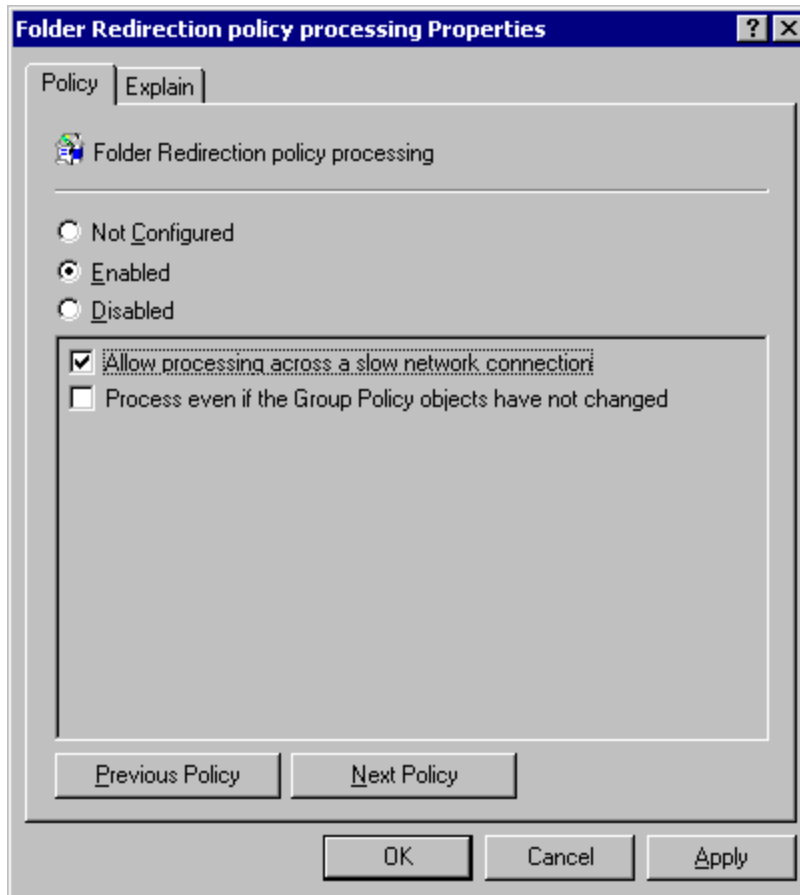


Figure 4.13: Enabling Folder Redirection across a slow network connection.

If you want to change what Win2K considers to be a slow network connection for machines subject to this policy, you can modify Group Policy Slow Link Detection to specify a link speed greater or less than the default 500Kbps. Choosing a larger number—such as 700Kbps—tells the machines to consider any link slower than 700Kbps to be a slow link. When the machines detect a link slower than 700Kbps, they won't process any policies that are set not to process. Specifying a lower number—say, 60Kbps—tells the machines that, unless their detected link speed is below 60Kbps, they should consider the link a fast link and process all GPOs normally.

For more information on slow-link detection in Win2K, check out the following Microsoft Support article: <http://support.microsoft.com/support/kb/articles/Q227/2/60.ASP>.

The Apply Group Policy for Computers Asynchronously during Startup and Apply Group Policy for Users Asynchronously during Logon policy items control whether asynchronous policy processing happens.

If you decide to modify the default behavior of each of these processing options, make sure that the GPO that controls the behavior is one that is centrally managed. This GPO might be a good candidate for control by your forest-wide administrators. You might also decide that you only want to link this GPO to selected OUs or, more appropriately for a policy like slow-link detection, to particular AD sites.

In addition, keep in mind that processing options like slow-link detection are put in place for a reason. If you have a Software Installation policy that delivers Microsoft Office to a set of users and you override the slow-link behavior for Software Installation, users dialing in to your corporate network on 56-Kbps modems may suffer an interminable wait as many megabytes' worth of Office try to install.

GPO Loopback Processing

Finally, I want to say a word about a feature called *GPO loopback processing*. I'll discuss this feature in greater detail in Chapter 5, but for now, I want to introduce the concept. Consider the problem where you have a special kind of workstation in your AD infrastructure. It may be a kiosk or other single-function device that is meant to run one application and one application alone. If a user defined in your AD environment walks up to that workstation, he or she will receive any machine-specific GPOs defined for that device as well as any user-specific GPOs defined for his or her user account. However, an administrator may not want the user to receive the normal user-specific GPOs on this particular device.

To handle this situation, Microsoft has developed loopback processing. (See the User Group Policy Loopback Processing Mode policy item in Figure 4.12 above.) As I said, I'll go into more detail about this in Chapter 5, so suffice it to say that loopback processing lets you "disable" a user's normal policy processing in favor of any defined on the GPO(s) that apply to the device he or she's logging on to. It's a special kind of user-policy override meant especially for machines with unique application requirements.

Summary

In this chapter, I discussed the challenges and decision points for designing and deploying GPOs in the enterprise. The key themes here are to start slow and keep it simple—using the features you're familiar with, such as Administrative Templates—before going full bore into the complex stuff. That said, you'll still want to ensure that part of your GPO rollout plan includes good version and change control, just as you would with any other infrastructure or application. Unfortunately, with no tools built in to Win2K to perform these tasks, you'll have to go it alone until third parties catch up and provide this functionality.

Finally, you'll need to consider the inherent special behavior that GPOs provide, like background, asynchronous, and slow-link detection processing. You can modify this behavior if necessary, but you'll need to balance that against good user experience and good GPO processing performance.