

Chapter 5: Working around Group Policy Limitations

In the preceding four chapters, I described Group Policy Object (GPO) function and structure and how you can make practical use of these functions in your own environment. The last two chapters of this book are directed at those of you who've actually gone so far as to implement GPO technology and are now coming up against its limitations. In particular, this chapter will discuss how to work around some limitations and challenges I've discovered while working with GPOs—particularly GPO delegation. In Chapter 6, I'll describe some techniques I've come up with for troubleshooting GPO problems and tools for managing GPOs before problems arise.

I think it's reasonable to expect that with any new software technology, the first implementation won't have every capability or anticipate the end user's every need. This is of course true with Microsoft's Group Policy implementation. This isn't to say that Microsoft hasn't done a yeoman's job with this first version. But as with most software, real-life user problems are difficult to anticipate and code for the first time around. That said, I'll look at some of the thornier aspects of GPO implementation and describe ways that I've found to get around them.

Controlling GPO Delegation

Why is a seemingly basic GPO function discussed in this chapter on Group Policy limitations? The answer is simple. In Chapter 4, I alluded to the challenges you can face when delegating administration of GPOs. These challenges make delegation of GPOs counterintuitive and difficult to do correctly. This section seeks to clear up some of these difficulties and help you manage GPO delegation effectively.

To review, you're going to be interested in delegating to your administrators three GPO tasks.

- **Creating**—Who is allowed to create GPOs and delete them in your Active Directory (AD) infrastructure?
- **Linking**—Linking (and un-linking) a GPO to a container—be it a site, domain, or Organizational Unit (OU). (By default, when you use a tool like AD Users and Computers or AD Sites and Services to create a new GPO, creating and linking occur in one step, but they're actually two distinct steps.)
- **Editing**—Once you've created and linked a GPO, you need to edit it.



In Chapter 3, I described how you can create a GPO without actually linking it to a container. See the tip in the section Adding New GP Links.

As you know, AD delegation is all about granting security rights to AD objects to permit certain users access to specific objects and their properties. You carry out GPO delegation using the same technique—by granting rights to certain AD objects related to GPOs (such as the Group Policy Container, or GPC). The principal tool for controlling GPO delegation is your old friend, the ACL editor.

As I mentioned in Chapter 4, to display the ACL editor, right-click the GPO display name in a Group Policy MMC snap-in, then choose Properties from the shortcut menu. In the Properties dialog box, click the Security tab. Figure 5.1 shows the ACL editor, focused on a GPO.

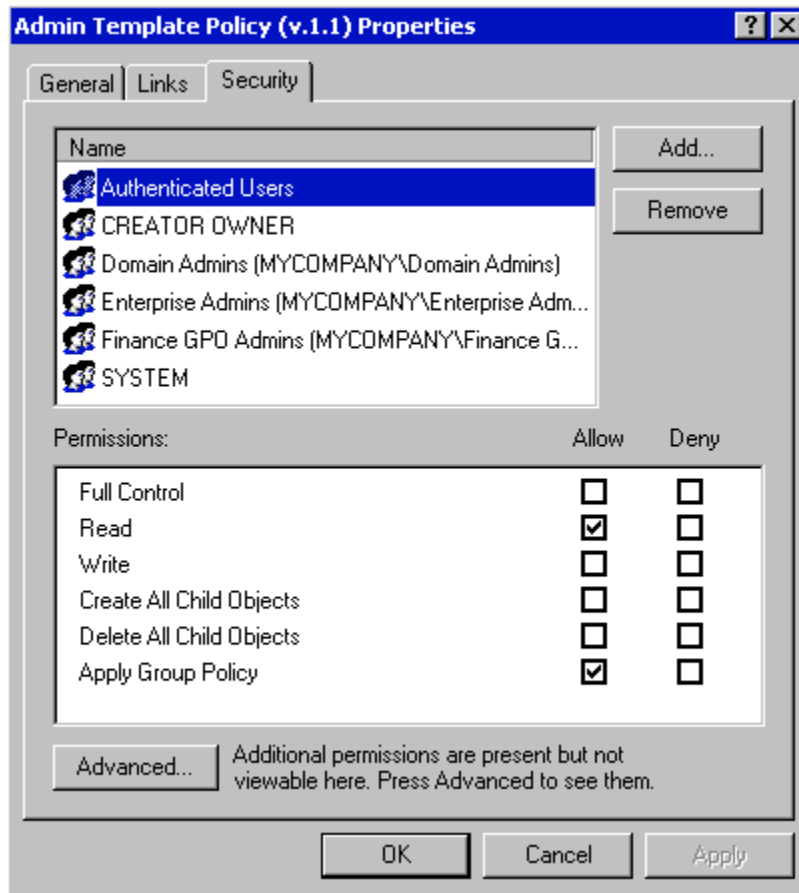


Figure 5.1: The ACL editor, focused on a GPO.

The ACL editor for GPOs has a dual function. You use it to filter security groups and thereby control which users and computers process a particular GPO. However, for the purposes of this discussion, you use it to control who can create, edit, and link to a particular GPO. When you delegate creating and editing of GPOs, you set security on the GPO itself (as in Figure 5.1). When you delegate linking GPOs to container objects like sites, domains, and OUs, you grant rights on the container object itself, as you might expect.

Default Rights

When you install Windows 2000 (Win2K) and AD out of the box, a certain set of default rights is granted on AD. As they affect the three roles I discussed above, these default rights are shown in Table 5.1 below.

This Group	Has the Ability to Do This
Enterprise Admins (global group or universal group in native mode)	Create, delete, and edit GPOs and link (and un-link) GPOs in all sites, domains, and OUs in a forest.
Domain Admins (global group)	Create, delete, and edit GPOs and link (and un-link) GPOs in the domain and in all OUs (but not sites) in which the Domain Admins group resides. Members of the Domain Admins group in the forest root domain can

This Group	Has the Ability to Do This
	also link (and un-link) to sites in a forest.
Group Policy Creator Owners (global group)	Can create (but not delete) GPOs in the domain to which the group belongs. The user who is a member of Group Policy Creator Owners can also edit any GPOs that he or she creates, but other members of that group cannot. In addition, members of this group cannot link to any site, domain, or OU.
Administrators (built-in local group)	Can create GPOs in the domain to which the group belongs. Can only edit or delete GPOs that were created by members of this group. Can link and un-link GPOs at the domain and on all OUs.

Table 5.1: The default AD rights that control GPO delegation.

As you can see from Table 5.1, the default GPO delegation configuration in AD is less than straightforward. No one group with the exception of the forest-wide and all-powerful Enterprise Admins group can create, delete, edit, link, or un-link GPOs universally in a forest. However, if you're trying to design a GPO delegation model that grants limited capabilities to your administrators, you're not going to want to make these administrators members of the Enterprise Admins group (or even Domain Admins or local Administrators) because these groups confer other, power capabilities as well. This leaves you using the Group Policy Creator Owners group (for brevity, I'll call this the GPCO group from now on) as a way of delegating GPO creation, then coming up with other solutions to take care of the other roles. As you can see, if you want to build a decent GPO delegation model, you have to modify AD security.

There are several approaches that you can take to solve the GPO delegation problem. The first is to modify the rights of one of the existing groups above. But because all but the GPCO group confer many other rights to their members, it's probably best to start from scratch and create a new group. Before I describe such an approach, I want to talk about the problems associated with using the GPCO group to delegate GPO management.

As I mentioned, the GPCO group only grants a member the ability to create GPOs in a domain; it doesn't allow the member to link these GPOs to any container. If user Joe Admin is a member of the GPCO group and he creates a GPO, he alone can edit this GPO by default. The reason is that when the Group Policy MMC snap-in creates the GPO in the domain, it doesn't assign the GPCO group as having the rights to the GPO. Rather, GPCO is a special kind of group that is just a placeholder for the actual user who is a member of the group and performing the operation.

When rights are assigned based on an operation performed by a member of the GPCO group, the Access Control Entry (ACE) is actually assigned to the user's account rather than to the group. As an example, let's suppose you have a user id in your AD domain called admDarren. The administrator for your domain has placed this user account in the GPCO group. You want to create a GPO, so you display the Group Policy MMC snap-in, and using the method described in Chapter 3, you create a GPO without linking it to a container (because members of the GPCO group cannot, by default, link GPOs to any container). Figure 5.2 shows what the ACE looks like on the newly created GPO when you view the security properties on the GPO.

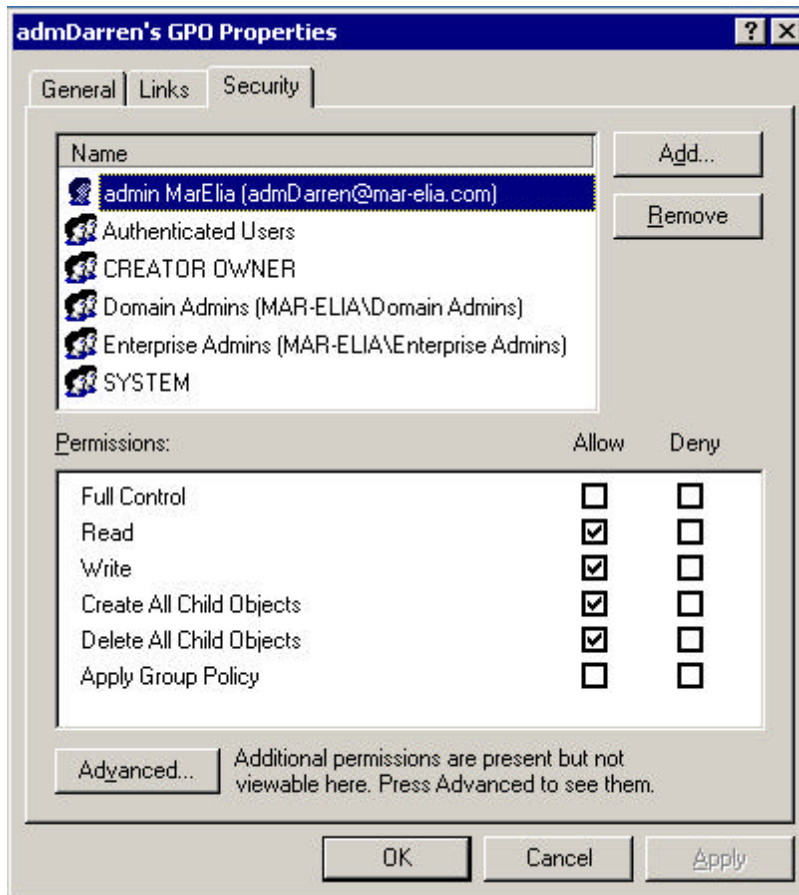


Figure 5.2: The security properties on a GPO created by a member of the GPCO group.

As you can see in Figure 5.2, none of the ACEs list the GPCO group. This is because the user id—admDarren—replaced the GPCO group in the ACE when you created the GPO. Why is this a problem? Well, once you create a GPO, you may want others to be able to edit it. Sounds reasonable, doesn't it? One would think that if a member of the GPCO group creates the GPO, another member of the group can edit it. Unfortunately, it's not as simple as that. The group only acts as a placeholder for the real user who is doing the creating. As you can see from the ACL in Figure 5.2, this account (in addition to the other groups), not the GPCO group, can modify the GPO.

So how do you get around this? To provide an answer, I need to explain where and how GPO delegation occurs in AD, and to do that, you need to recall concepts from Chapter 2 on the structure of GPOs.

How GPO Delegation Works

As I've mentioned previously, two sets of tasks can be delegated as part of GPO management. The first set of tasks is creating, editing, and deleting a GPO in the domain. The second is linking the GPO to a site, domain, or OU.

Delegating Creation, Editing, and Deletion GPOs

Remember from Chapter 2 that GPOs “belong” to a particular domain and are broken into two physical pieces—the Group Policy Container (GPC) and the Group Policy Template (GPT). The GPC is the part of a GPO that resides in AD. GPC objects can be found in the AD Users and Computers MMC snap-in (in Advanced View mode) under the System\Policies folder (see Figure 5.3).

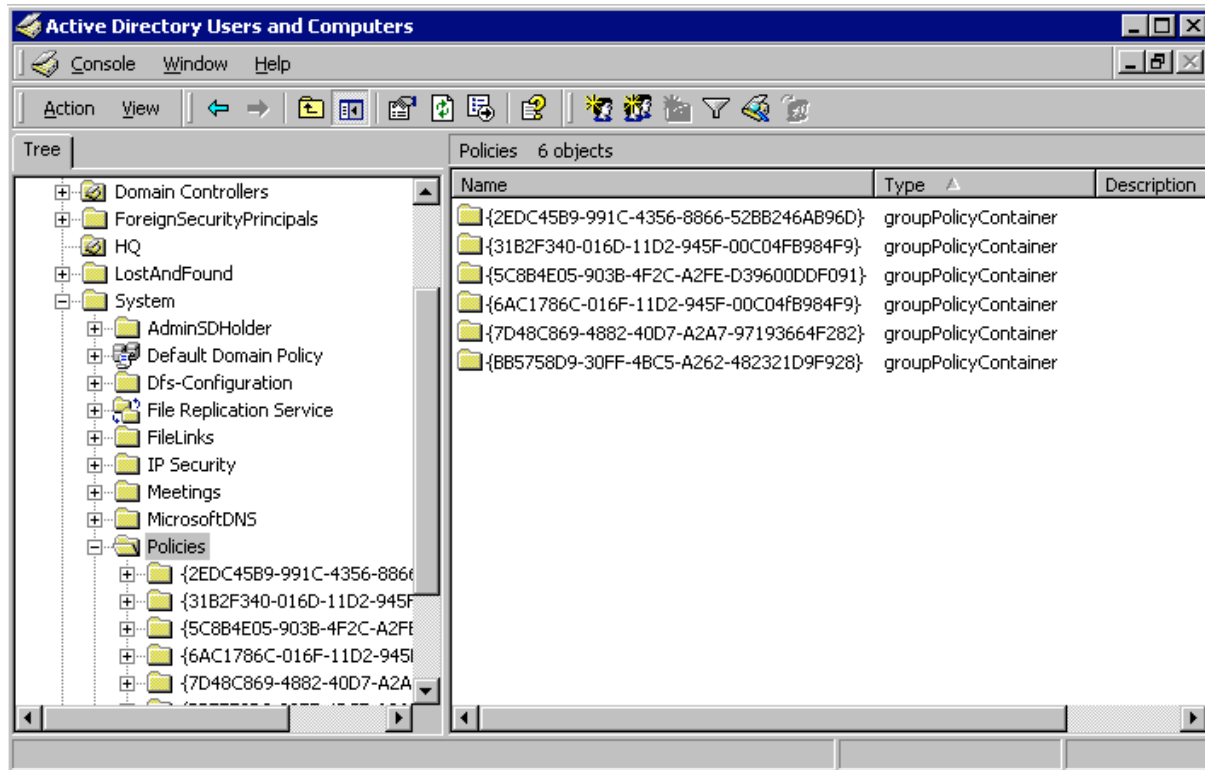


Figure 5.3: The GPC portion of a GPO, displayed in the AD Users and Computers MMC snap-in.

Figure 5.3 shows a number of Globally Unique ID (GUID)-named folders under the System\Policies folder that correspond to GPOs created in this domain. Just as with other objects in AD, these folders have security rights associated with them. These rights control who can do what to a particular GPO. Figure 5.4 shows the ACL editor for one of these folders.

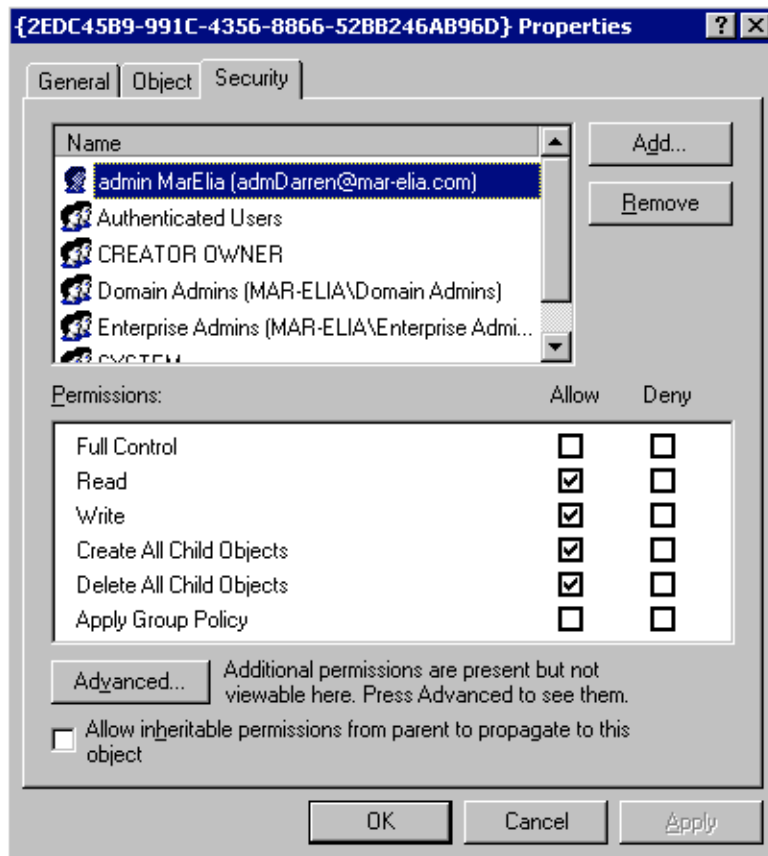


Figure 5.4: The security properties of a GPC in the AD Users and Computers MMC snap-in.

You'll notice that the permissions shown in Figure 5.4 are identical to those listed in Figure 5.2. In fact, they're for the same GPO—admDarren's GPO. Figure 5.2 showed the security settings on the GPO using the Properties command on the shortcut menu in the Group Policy MMC snap-in, whereas Figure 5.4 shows the security settings on the GUID-named GPC folder corresponding to that GPO.

As you can see, Figure 5.2 shows a "sanitized" version of the rights assigned to the GPC. Furthermore, if you were to look at the file permissions on the GPT for this GPO in the SYSVOL share in this domain, you'd see file security like the AD security shown above. Figure 5.5 shows the ACL editor focused on the GPT for this GPO.

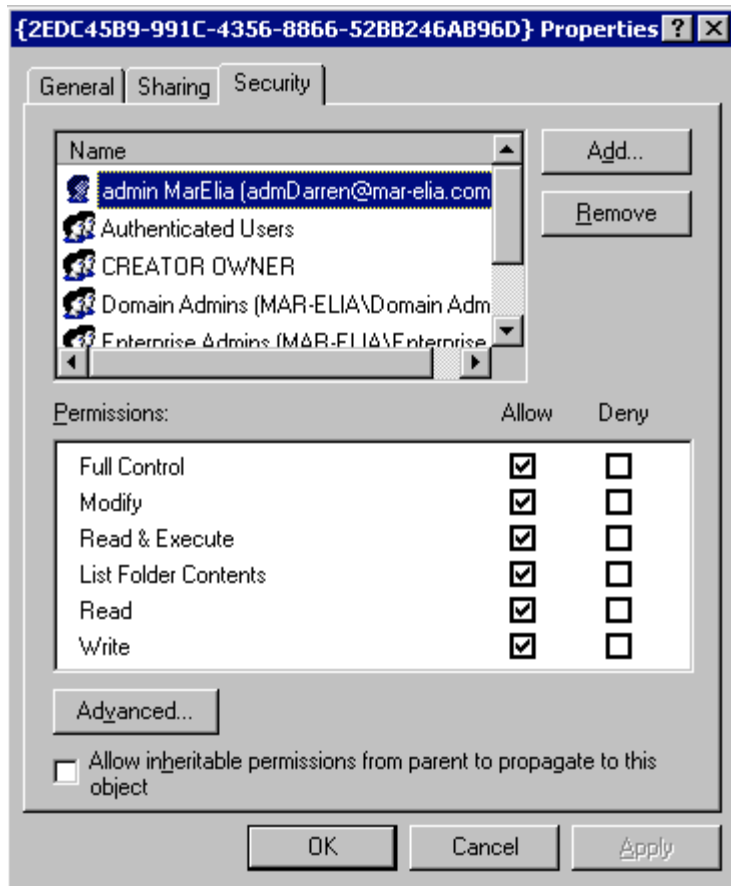


Figure 5.5: The NTFS security rights on a GPT in SYSVOL.

Of course, because NTFS rights are a bit different from AD rights, figures 5.4 and 5.5 don't list identical ACEs, but the same users and groups are listed in the ACL, and the control they're granted over the GPT is similar to the control granted in the GPC.

It isn't mere coincidence that the GPC and GPT have nearly the same rights assigned to them for a particular GPO. When you create a GPO using the Group Policy MMC snap-in, several things go on behind the scenes to ensure that the GPO has the correct security assigned to it. The security rights granted to a new GPO are controlled based on simple inheritance. In the case of the GPC, when a new GPO is created, a new GUID-named folder is created, like those shown in Figure 5.3. This folder inherits the permissions granted to its parent folder, System\Policies.

The rights granted at the folder level control who in a domain can create, edit, and delete a GPO. Similarly, on the GPT side, the same users or groups who have rights to create, edit, and delete GPC objects are granted such rights over the GPT's parent folder. Because the GPT resides in the SYSVOL share on domain controllers in a domain, this parent folder is found on the path `\\<domain>\SYSVOL\<domain>\policies`.

When you view the security properties on a GPO from a Group Policy MMC snap-in (as shown in Figure 5.2 above), you're viewing the security as it's assigned on the GPC folder for that GPO. But to be able to create, delete, or edit the GPO, you also need to have the corresponding rights in the GPT folder for the GPO. In a default AD installation, if you keep the default groups

that have permissions to manage GPOs, you don't ever need to be concerned about the security permissions applied to either the GPC or the GPT Policies folder. However, if you want to do any customization to GPO delegation, you need to modify these default permissions on the GPC or GPT.

 You may be inclined to ask why the Delegation of Control (DoC) wizard provided with the AD Users and Computers and AD Sites and Services snap-ins doesn't allow you to customize the delegation of GPO creating, deleting, and editing. It's a good question, and one with no easy answer. The DoC wizard is essentially a shortcut to the ACL editor, so it's a simple and rather stupid tool. If you run the DoC wizard against a container such as a site, domain, or OU, there appears to be a way to delegate creating GPC objects. However, this is in fact not what occurs.

Because GPC objects are created under the special System\Policies folder in an AD domain, you actually need to run the DoC wizard on that folder rather than on a particular site, domain, or OU. Makes sense, right? However, simply running the DoC wizard against the System\Policies container doesn't grant the rights needed to delegate creating GPOs. This is because the DoC wizard does nothing to handle creating permissions in the GPT in SYSVOL. As a result, if you use the DoC wizard to delegate the creating of GPOs to a new group, then as a member of that group, try to create a GPO in the domain, you'll get an "Access Denied" message. Later in this section, I'll describe the methods you can use to successfully delegate creating, deleting, and editing of GPOs.

The first step to correctly delegate GPO administration is knowing which rights confer which capabilities. Let's look at exactly what rights you need on the GPC and GPT to be able to view, edit, create, and delete GPOs. Table 5.2 below shows the rights that must be granted on the System\Policies folder or a GPC to confer a particular GPO management capability. (These rights correspond to the ACL editor view shown in figures 5.2 and 5.4.) The Apply Group Policy permission isn't listed because it doesn't affect delegation of GPOs.

This Right	Allows the User or Group To
Full Control	View, create, delete, and edit the GPO.
Read	View the GPO and its properties in the Group Policy MMC snap-in but not open the GPO for editing.
Write	View and edit the GPO. (Write doesn't confer the ability to read a GPO as well; that permission must be explicitly granted.)
Create All Child Objects	Create and edit (but not delete) GPOs.
Delete All Child Objects	Delete GPOs.

Table 5.2: The rights used on GPC objects to delegate GPO management.

Table 5.3 below lists the corresponding NTFS rights that must be set on the GPT folders to grant capabilities like those listed in Table 5.2 above.

This Right	Allows the User or Group To
Full Control	View, create, delete, and edit the GPO.
Modify	View, create, delete, and edit the GPO.
Read & Execute List Folder Contents Read	View the GPO. When you select the Read & Execute right, the List Folder Contents and Read rights are granted automatically.

Write	View, create, delete, and edit the GPO.
-------	---

Table 5.3: The rights used on GPT objects to delegate GPO management.

 It's possible to confer one set of rights on the GPC portion of a GPO and another set on the GPT portion. The results, as you can imagine, aren't good. For example, you can grant Write access to a GPO in the GPC. This theoretically lets you edit the GPO, and indeed, you can open and edit the GPO just fine in the Group Policy MMC snap-in. However, if you don't have the corresponding Write permission on the GPT portion of the GPO, any changes you make to the GPO are lost without any kind of warning or message about a lack of access. It seems that the Group Policy MMC snap-in only checks to ensure that you have the correct permissions on the GPC before allowing you to edit that GPO.

Of course, creating and editing GPOs is only one side of the delegation equation. The other side is linking and un-linking GPOs from a site, domain, or OU.

Linking GPOs

Once you understand the delegation process for creating, deleting, and editing GPOs in your AD domains, the next step is to decide who can link these GPOs to sites, domains, and OUs. As I mentioned above, while delegating control of tasks such as creating GPOs requires setting security rights on the GPC and GPT, delegating the linking and un-linking of GPOs is done at the container level.

For example, if you want a group of administrators in the Finance OU to be able to link GPOs to its own OU, you need to set security rights on the OU rather than on the GPC or GPT for the GPOs. Fortunately, unlike delegating the creating of GPOs, where you have to worry about security on both AD and SYSVOL, delegating GPO linking only involves setting AD security on the container objects in question.

There are actually two rights that you can grant to a user or group at the container level. Remember from Chapter 3 the discussion of the `gPLink` and `gPOptions` attributes that are present on container objects. The `gPLink` attribute contains the GUIDs of all GPOs linked to the selected container. Figure 5.6 shows the contents of the `gPLink` attribute on the HQ OU in a test domain using the ADSI Edit tool.

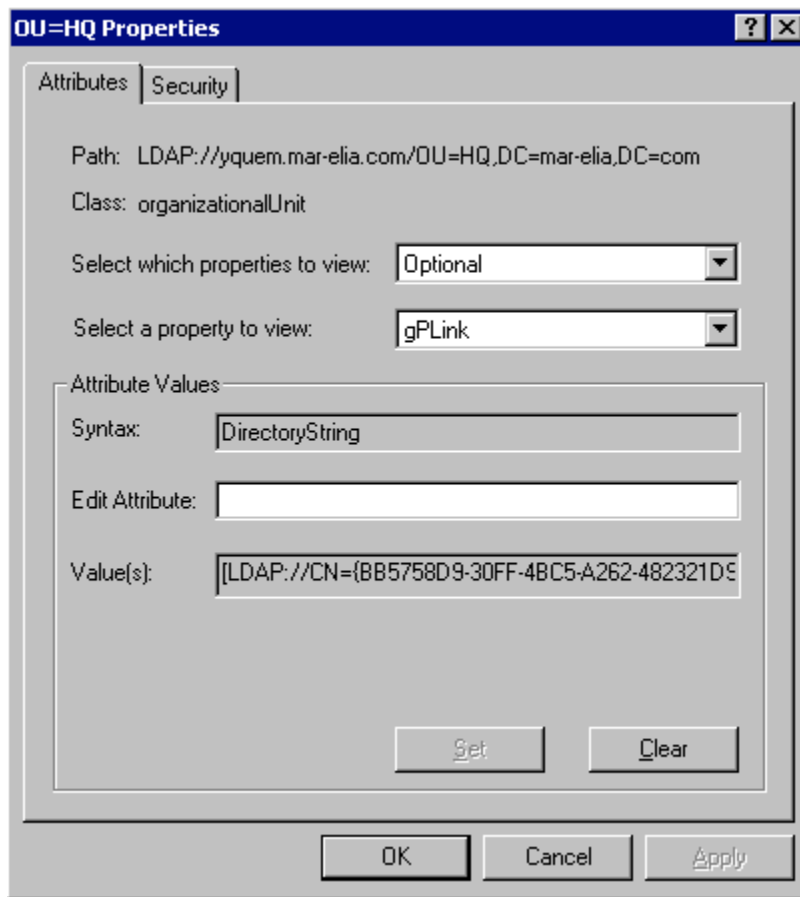


Figure 5.6: The contents of the *gPLink* attribute on an OU in ADSI Edit.

The Value(s) text box shows a string containing the Lightweight Directory Access Protocol (LDAP) distinguished name of a GPO in the domain. The ability to write to the *gPLink* attribute on a given site, domain, or OU container object is what gives a user or group the ability to link and un-link GPOs to that container.

In addition to being able to manipulate the *gPLink* attribute, you can delegate the ability for a user or group to use the Block Policy Inheritance feature, described in Chapter 3. The *gPOptions* attribute on a container object controls whether this feature can be used. If the user or group can write to this attribute, they can enable or disable this feature. If they cannot write to this attribute, they cannot change it. Thus, even if you allow someone to link a GPO to their OU, for example, you can prevent him or her from disabling “upstream” GPOs by not granting sufficient permissions on the *gPOptions* attribute.

There are two ways to delegate security rights (and thus be able to link and un-link GPOs) for the *gPLink* and *gPOptions* attributes. You can use:

- The DoC wizard to grant rights to users and groups in your domain
- The ACL editor directly.

I’ll explain how to delegate security rights using the ACL editor. The DoC wizard is only good for one-time ACL changes, and eventually, you’ll have to use the ACL editor anyway.

First start the AD Users and Computers MMC snap-in. (If you're granting rights to users or groups on site objects, start the AD Sites and Services snap-in.) Let's suppose you have an OU called HQ, and you want to grant linking and un-linking rights to an administrative group in this OU called HQ Group Policy Admins. The first thing you need to do is get to the ACL editor for the HQ OU. Right-click the HQ OU, then choose Properties from the shortcut menu. In the Properties dialog box, click the Security tab. Figure 5.7 shows the ACL editor for this OU.

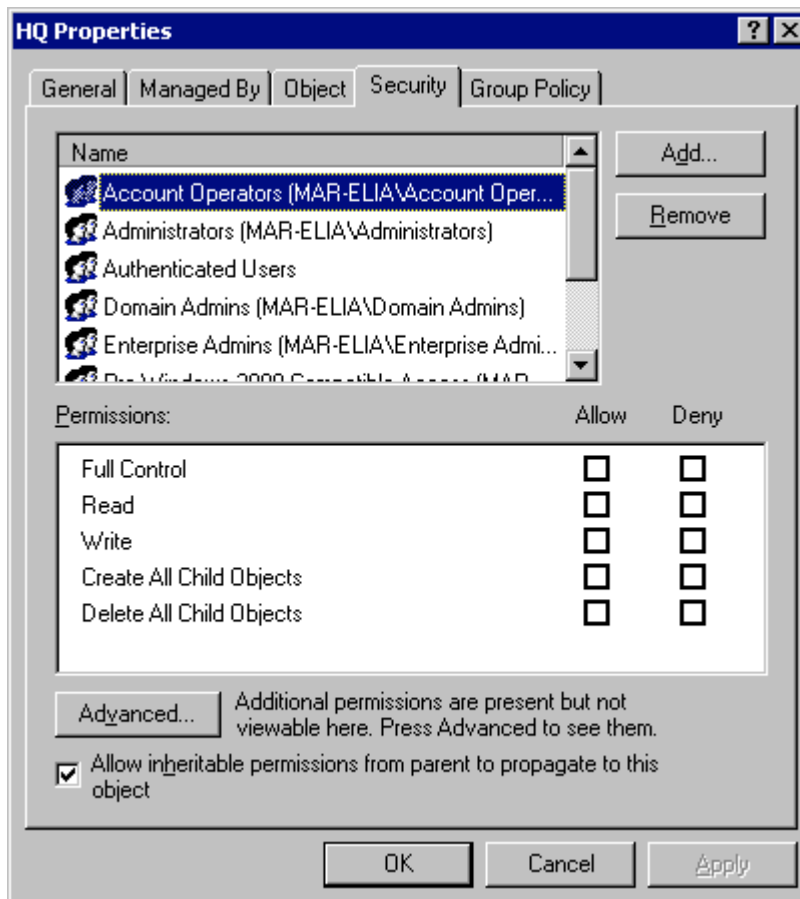


Figure 5.7: The ACL Editor, focused on the HQ OU.

To grant (or revoke) rights for the gPLink and gPOptions attributes on this OU, click Advanced to see additional permissions. Figure 5.8 shows the advanced ACL editor view.

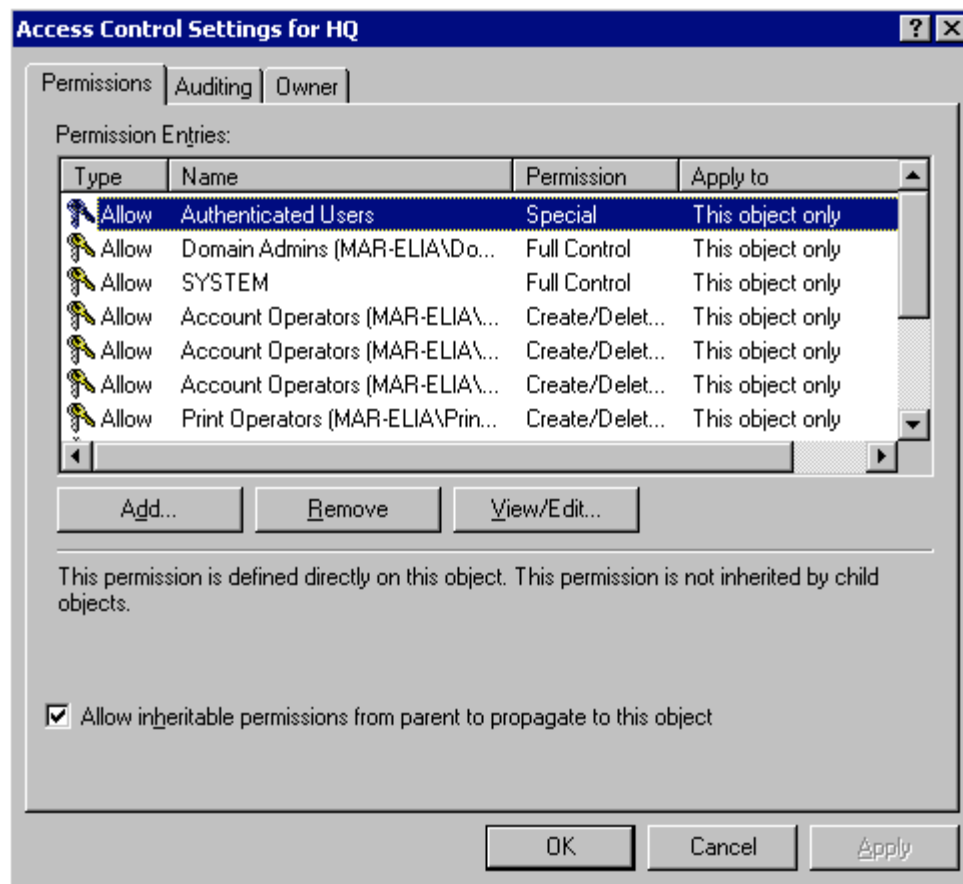


Figure 5.8: The advanced ACL editor view on an OU.

To delegate the linking and disable inheritance rights on this OU, click Add, then add the HQ Group Policy Admins group to the ACL. Select the group, and the Permission Entry dialog box appears, where you can assign the actual rights to the gPLink and gPOptions attributes. This dialog box has two tabs—Object and Properties. Object is selected by default and shows the permissions available on the OU object itself. However, you're interested in the Properties tab. Click this tab, then set permissions on the individual properties on this OU. Figure 5.9 shows the Properties tab as well as the permissions for gPLink and gPOptions.

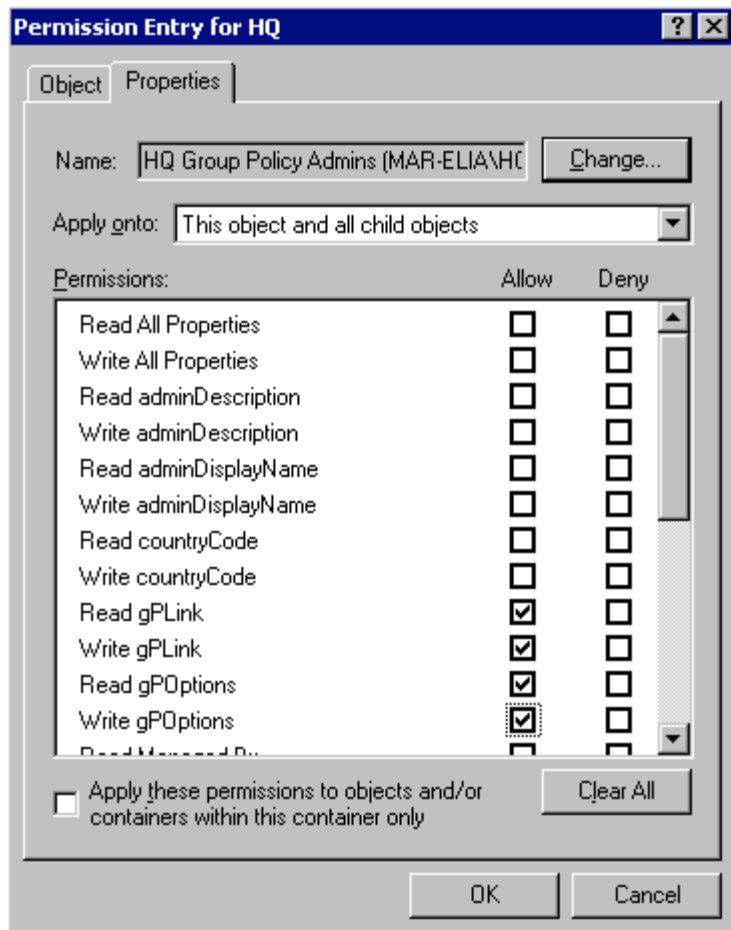


Figure 5.9: The permissions on properties (attributes) that you can assign on an OU.

As you can see in Figure 5.9, both gPLink and gPOptions have Read and Write permissions associated with them. Read permissions are required for a user or computer processing GPOs because it needs to be able to read the gPLink attribute to see which GPOs are linked to the site, domain, or OU in question. However, the Write permission is of more interest here. By selecting the Write gPLink permission, you grant the HQ Group Policy Admins group the ability to link GPOs to the HQ OU. And by selecting the Write gPOptions permission, you grant this group the ability to enable or disable inheritance on this OU.

How will you know if you've successfully granted or revoked from an administrator the ability to link or un-link GPOs to or from a container object? When the administrator hasn't been granted the Write gPLink permission, then attempts to create a GPO or add a link to an existing GPO from a container object, that option will be grayed out.

You can see this, for example, in the AD Users and Computers snap-in. If you right-click HQ OU, choose Properties from the shortcut menu, then click the Group Policy tab in the Properties dialog box, you'll see the familiar Group Policy dialog box (shown in Figure 5.10), where the New, Add, Options, and Delete buttons are all grayed out. Removing the Write permission on gPLink effectively prevents this user from adding any new GPOs or linking existing ones to this

OU. And because the user can't change the gPLink attribute, he or she also can't delete a GPO or its link from this dialog box.



Figure 5.10: The Group Policy dialog box after the gPLink Write permission has been revoked.

Interestingly enough, the Block Policy Inheritance check box is also grayed out, even though this user still has Read and Write permissions for the gPOptions attribute. Now, I personally think that users should still be able to block policy inheritance in this situation, but apparently the person at Microsoft who coded this user interface (UI) didn't think so! Another seemingly odd thing is that if a user has Write permission on gPLink, but not on gPOptions, he or she can select the Block Policy Inheritance check box, but then receives an error, indicating that the change can't be made. You'd think that the better way to handle this would be to gray out the check box, as shown in Figure 5.10 above.

Now that I've discussed how you can delegate GPO creating, editing, deleting, and linking, it's time to pull it all together and go through a real scenario where you delegate OU administration in your AD domain.

Putting GPO Delegation into Practice

Let's suppose that you've installed your AD domain, and now you want to use your newfound knowledge about GPO delegation to delegate the management of GPOs in your domain. The first

thing you'll do is delegate the ability to create, edit, and delete GPOs. Remember from the discussions so far that these are domain-wide rights. That is, a GPO exists in a domain in a single place—in the GPC (and GPT). So when you delegate administration of GPO creation, deletion, and editing, you do so across the domain. If you grant users the ability to create GPOs in a domain, they can create as many as they want until you stop them. On the other hand, if you grant users the ability to link a GPO to the Finance OU, they're fairly limited to how much damage they can do.

The point is that delegating GPO creation and deletion is pretty much an all-or-nothing task, one that should be done with prudence. Because each new GPO creates its corresponding GPC and GPT, which starts out at just under 1 megabyte (MB) in size, a user who creates thousands of GPOs just for the fun of it can seriously affect seemingly unrelated things, such as AD and NT File Replication Service (NTFRS) replication traffic on your network. So consider GPO creation (and deletion) a rare right that shouldn't be abused!

That said, you do have some ability, once a GPO has been created, to go back and remove permissions to prevent a user from deleting it or even editing it. I'll talk in this section about how to delegate the creation process from scratch as well as how to change existing permissions on a GPO to prevent further changes.

Making Changes to the Policies Folder

As I mentioned in the previous section, delegating the creation of GPOs is controlled using the permissions contained in the GPC System\Policies folder in an AD domain and the NTFS file permissions on the SYSVOL-based Policies folder on your domain controllers. Therefore, if you want to grant new rights to create GPOs, you need to make changes to these folders.



To reiterate, you only have to manually modify GPC and GPT permissions when you're delegating the **creation** of GPOs to a new user group. To grant users the ability to modify or delete a GPO, you can use the ACL editor.

Let's suppose you've created a special domain-wide global group called GPO Admins. You want to be able to give this group the ability to create, delete, and edit GPOs in your domain. You'll then delegate editing and linking rights on an as-needed basis to those groups that require them. I'll walk through this entire scenario now to describe the challenges and complexities of implementing a truly flexible GPO delegation.



Depending on the size of your environment, you may make some groups responsible for creating, editing, and deleting GPOs and delegate to others the ability to link GPOs. This is important to keep in mind as you plan your GPO implementation because the various groups dealing with GPOs must be in sync at all times.

Open the AD Users and Computers MMC snap-in, focused on your domain—specifically, the domain controller in your domain that is currently the primary domain controller (PDC) operations master)—and drill down to the System\Policies folder.



To focus on a particular domain controller in the AD Users and Computers snap-in, right-click the domain name at the top of the left-hand (scope) pane, then choose Connect to Domain Controller. In the Change To dialog box, type the name of the domain controller that has the PDC operations master role for your domain. The reason I suggest the PDC operations master to focus on for your GPC changes is that it's the domain controller that the Group Policy snap-in focuses on by default whenever you edit the GPO. Focusing on the same domain controller each time you make changes to a GPO (whether using the MMC snap-in or directly on the GPC or GPT) ensures that two people don't edit the same GPO on two different domain controllers and potentially overwrite each other's changes.

Right-click this folder and choose Properties from the shortcut menu. Then in the Properties dialog box, click the Security tab. When you open the ACL editor on this folder, you see the default GPO permissions that are listed in Table 5.1 earlier in this chapter. At this point, you're ready to grant create, edit, and delete permissions to the GPO Admins group.

Click Add, then select the GPO Admins from the list. Now that you've added an ACE for the group, you need to modify the permissions you're granting to it. How do you know which permissions to grant to a group to convey sufficient results? Well, from Table 5.2 earlier in this chapter, you know that you need to grant at least the Create All Child Objects and Delete All Child Objects permissions to give the group the ability to create and delete GPOs.



If you don't grant the Delete All Child Objects permission, the group members can create but not delete GPOs in the domain.

You can also grant Read permission, but you don't need to, because the Authenticated Users group, of which all users defined in the domain are members, already has Read access to the System\Policies folder by default. Figure 5.11 shows the necessary permissions to grant the GPO Admins group the ability to create, edit, and delete GPOs in your domain.

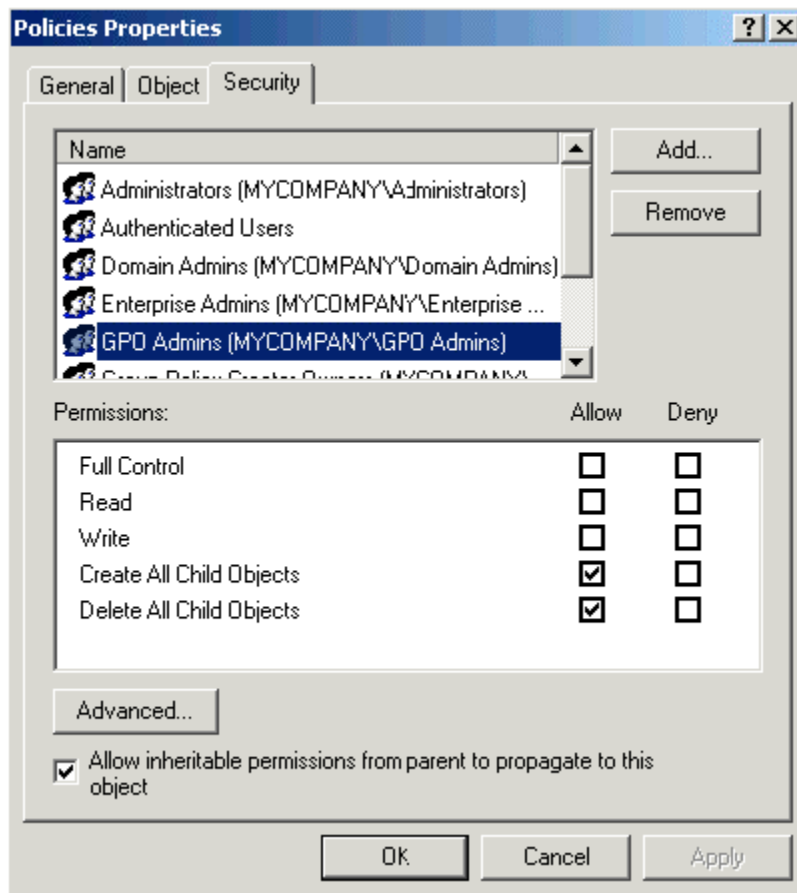


Figure 5.11: The permissions on the GPC-based System\Policies folder required to delegate creating, deleting, and editing of GPOs.

At this point, you're only half done. To really allow the GPO Admins group to be able to create, delete, and edit GPOs, you also have to grant the equivalent rights in the GPT. The easiest way to do this is to log on to a domain controller in your AD domain and change the NTFS permissions on the folder.

 Be careful which domain controller you're focused on when you make the change. Just as with the System\Policies folder discussed above, it's a good idea to change GPT permissions on the PDC operations master for your domain because this is the server, by default, that the Group Policy MMC snap-in focuses on when creating, editing, and deleting GPOs.

Log on to the PDC operations master, navigate to the volume where you've defined SYSVOL on that server, then drill down to the Policies folder under the SYSVOL share. The path will look something like this:

`%systemroot%\sysvol\sysvol\<domain>`

where *<domain>* is the name of your domain. In the domain folder, there should be at least two folders—Policies and Scripts. If you right-click Policies, then choose Properties from the shortcut menu, you can set security on the GPT. In the Properties dialog box, click the Security

tab, then add the GPO Admins group and grant it the appropriate rights to create, edit, and delete GPOs. In this case, the Modify permission will accomplish your goal, so grant GPO Admins the Modify permission on the Policies folder (see Figure 5.12).

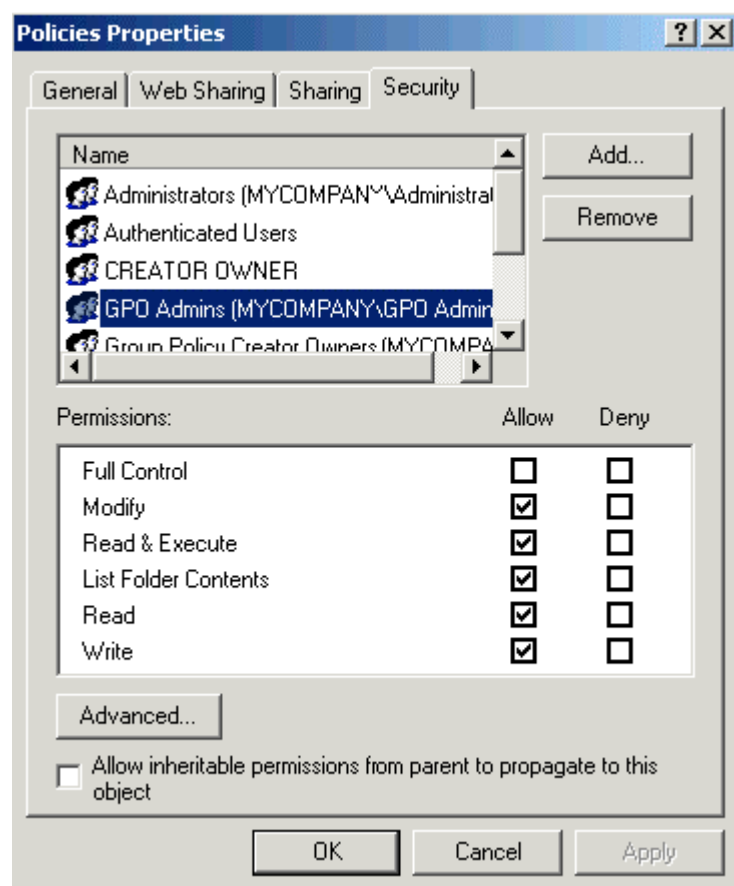


Figure 5.12: The permissions on the GPT-based Policies folder required to delegate creating, deleting, and editing GPOs.

You’ve now completely delegated creating, deleting, and editing GPOs to the GPO Admins group. The next step is to delegate linking to other user groups. Then I’ll show you how to modify existing GPO permissions to change access to a GPO.

Let’s say you have an OU called Finance that wants to be able to link to some GPOs in your domain. In the Finance OU is a group called Finance GPO Admins, to whom you’ll delegate the ability to link and un-link GPOs. As I mentioned earlier in this chapter (see “Linking GPOs”), you need to grant linking permissions on the container (for example, site, domain, or OU) where the linking is to occur. So as an administrator, you’ll grant linking permissions to the Finance GPO Admins group on the Finance OU.



The person who is changing security on the OU must have the rights to do this. In many cases, OU administrators themselves may be able to grant this permission. On the other hand, you may want a higher-level administrator to do this; you may not want to delegate the ability to change permissions on an OU to the administrator of the OU!

To delegate the linking permission, start in the AD Users and Computers snap-in and right-click the Finance OU. Choose Properties from the shortcut menu, then in the Properties dialog box, click the Security tab. In the familiar ACL editor, click Advanced. In the advanced ACL editor view, click Add to add the Finance GPO Admins group as the target group for the new ACE. The Permission Entry dialog box will appear. Click the Properties tab to set permissions on the OU's properties (attributes), then select the Read gPLink and Write gPLink permissions.

You can also grant the Finance GPO Admins group the ability to block policy inheritance on the OU by granting the Read gPOptions and Write gPOptions permissions. Figure 5.13 shows what the resulting ACE looks like after you've granted linking permissions on the Finance OU.

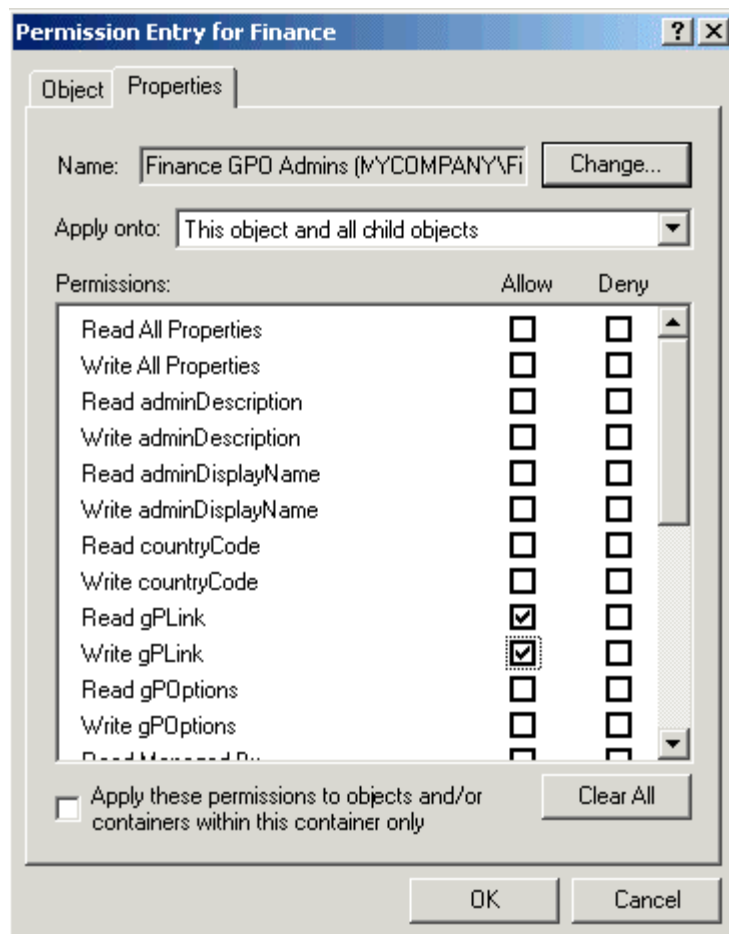


Figure 5.13: The permissions required to delegate linking and un-linking of GPOs.

Once you've granted these permissions, the Finance GPO Admins group will be able to link any GPOs that it has Read permissions on to the Finance OU. This is an important point. If there are

GPOs in your AD forest that you don't want to allow certain administrators to link to, you need to set permissions on those GPOs to prevent them from even being read. Remember that, by default, all GPOs are created with an ACE that allows the built-in Authenticated Users group Read and Apply Group Policy permissions. If there are particular GPOs that administrators shouldn't link to, you need to remove the ACE and replace it with a similar one that grants access only to the intended target group.

Remember, too, that once you remove this ACE, no users or computers will be able to read or process the GPO, so make sure that you replace it with one that grants the access you really need. An administrator needs at least the Read permission to be able to link to the GPO. As I mentioned in Chapter 3, the Apply Group Policy permission controls which users and computers actually process the policy, and an administrator doesn't have to be able to link to that GPO.

Modifying Permissions on Existing GPOs

As I alluded to above, you may need to modify the permissions on existing GPOs rather than modify permissions directly on the System\Policies and SYSVOL folders to delegate *new* GPO creation. If this is the case, I suggest using the Group Policy snap-in ACL editor instead of modifying permissions on each GPO's GPC and GPT folders. This minimizes mistakes and ensures that the permissions that you intended are indeed applied. I can think of a couple of examples where you might want to go back and modify existing GPO permissions.

In one situation, let's suppose you've decided to use the GPCO group to delegate creating and deleting GPOs. As I mentioned earlier in this chapter (see "Default Rights"), when you use this group, the ACE that is written to the newly created GPO is created for the user account that created the policy rather than the GPCO group. This means that, in the future, only that user can modify or delete the GPO. This may not be a desirable scenario, so it might be worthwhile to modify permissions on that GPO to give additional users or groups the Write or Modify permission. You can do this in the Group Policy MMC snap-in by focusing on that GPO and changing its security properties. (I discussed how to do this in "How GPO Delegation Works" earlier in this chapter.)

In another case, you might want to delegate administration of the two built-in GPOs—Default Domain Policy and Default Domain Controllers Policy. By viewing the properties of these two GPOs from MMC, you can determine their GUIDs, and you can modify the System\Policies and SYSVOL Policies folders to allow users other than Domain Admins, local Administrators, and Enterprise Admins to edit them.

Linking GPOs to a Site Object

Now I want to say a word about site-linked GPOs. Everything I've discussed in this chapter so far applies to creating and linking site GPOs, but you need to be aware of a few minor differences. The first and most obvious one is that to link GPOs to a site, you use the AD Sites and Services MMC snap-in. You simply right-click a site object in the snap-in, choose Properties from the shortcut menu. This is shown in Figure 5.14. Then in the Properties dialog box, click the Group Policy tab as you would in the AD Users and Computers tool.

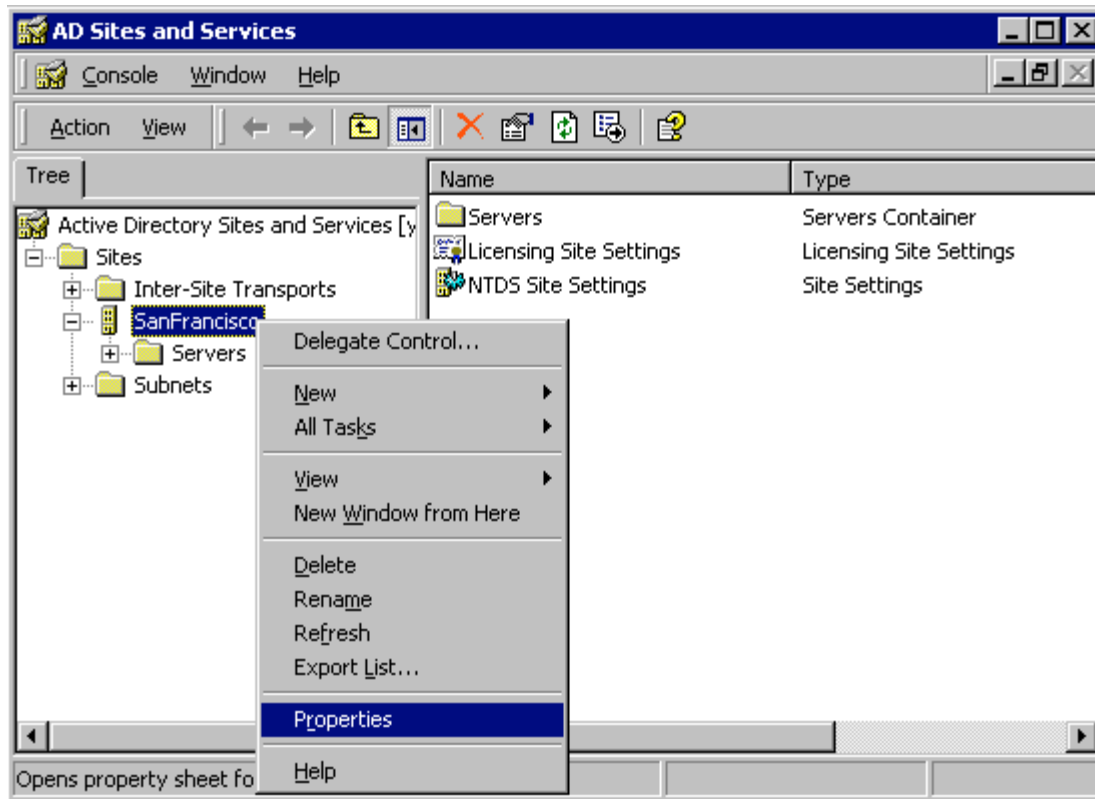


Figure 5.14: Linking GPOs to a site object in the AD Sites and Services snap-in.

The principal difference between site-based GPOs and domain- or OU-based GPOs is where they're created by default. If you're using the MMC snap-in to create and link a GPO to a site in one step, the GPO is automatically created in the forest root domain. If you have a multi-domain AD forest, you need to have GPO creation permissions on the GPC and GPT in the forest root domain to create and link site-based GPOs in one step. Otherwise, you can create a GPO, then link it in any domain in your forest to a site object.

Using Security-Group Filtering on GPOs

Now that I've discussed the challenges of controlling GPO delegation, I want to cover a related topic—using security-group filtering on GPOs. In previous chapters, I described how security-group filtering of a GPO prevents selected users and computers from processing it. In this chapter, I want to point out some challenges you need to be aware of when using security-group filtering.

As you can imagine, GPO security-group filtering can become complex in a hurry. If you have lots of GPOs at many levels in your AD infrastructure, security-group filtering isn't advisable. You need to trade off linking GPOs higher in your AD hierarchy and filtering their effects narrowly using security groups against linking at the OU level, right where your target users and computers are, and not filtering at all. In large, complex GPO implementations, I always recommend the latter if possible. However, you may find that security-group filtering is the only way you can control who gets a certain GPO. In that case, you need to be aware of a few points.

First, security is applied to the GPO, not the link! This is important. For example, if multiple OUs have been linked to a single GPO, the security that is applied on each of them is the same and is whatever has been set on the GPO (that is, the GPC and GPT). This can get confusing! If Joe Admin creates a GPO for desktop lockdown, then sets permissions so that only the Finance Users security group has Read and Apply Group Policy permissions on that GPO, no one else who links to the GPO may be able to process it unless they too modify the permissions. This is another good argument for having a centralized group control GPO creation and maintenance and allowing your OU administrators to simply link and un-link GPOs.

Second, remember that you have to first remove the Authenticated Users ACE from the default GPO permissions before you can effectively use security groups to filter the GPO's effects. This is because Authenticated Users includes every user and computer in your AD forest. You won't have any luck filtering a GPO if that ACE is still in place. Remember, however, that when you remove that ACE and add a new security group, only those users and computers that are part of the new group will be able to view and process the GPO.

Finally, be aware that security-group filtering applies to the whole GPO. For example, it might be nice if you could filter Folder Redirection for one set of users and Administrative Templates for another set of computers in a single GPO. No can do! However, you do have some granularity of control in the Software Installation feature in GPOs, and I'll talk about that next.

Security-Group Filtering for Applications

When I first encountered GPOs, I thought it'd be really nice to have more granular control of security-group filtering. That would allow me, in a single GPO, to control which users and computers processed which features. Unfortunately, for the most part, security-group filtering of a GPO is an all-or-nothing affair.

The good news is that Microsoft does provide more fine-grained control when you're using the Software Installation feature in a GPO to publish or assign applications to your users. If you publish or assign several applications in a single GPO, you can control which user or computer groups will see them. If they can't see them, the applications won't get installed. Software Installation is a kind of down-market version of the Microsoft Systems Management Server (SMS) user and computer group-based Software Advertisement feature.

 For an explanation of publishing and assignment, see chapters 1 and 3.

For example, suppose you have a domain-linked Software Installation GPO and you want to use it to deploy all of your applications using the Software Installation feature in Group Policy. However, you want to be able to control who receives certain applications in the GPO. You can do this by modifying permissions on the individual applications that you've published or assigned. Figure 5.15 shows a GPO with two applications deployed—Microsoft Office 2000 and Windows 2000 Administration Tools.

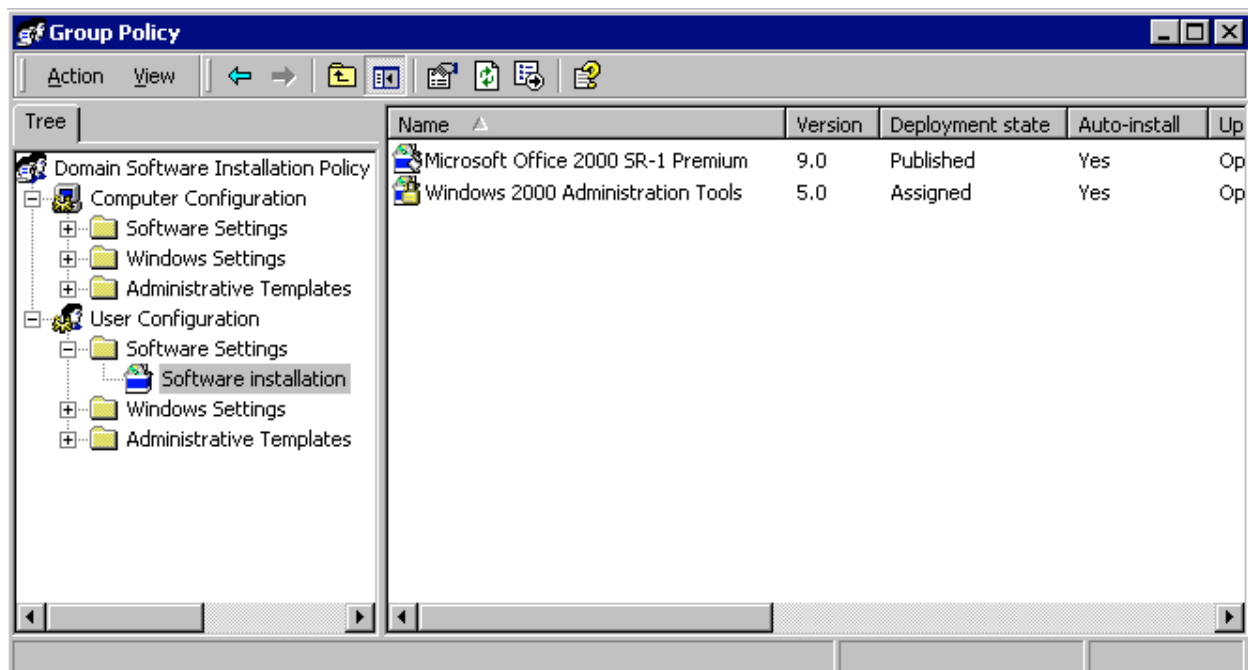


Figure 5.15: Applications deployed using the GPO-based Software Installation feature.

Let's say you want to give everyone access to Office 2000, but you want only certain members of an administrator group to have access to Win2K Admin Tools. Perhaps you've removed the Authenticated Users ACE—which grants all users and computers Read and Apply Group Policy permissions on a GPO—and replaced it with a security group that is targeted to specific users or computers. Now you want to set permissions on Win2K Admin Tools so that only members of a particular administrative group receive it.

Before the group members can read the permissions you assign to Win2K Admin Tools, they must have permission to process the GPO as a whole. Thus, before you use security-group filtering on an application, the GPO has to be visible and available for processing by all users (or computers).

That said, I'll show you how to set permissions on the two applications shown in Figure 5.15 to meet these requirements. The first thing you need to do is edit the GPO, so start the Group Policy MMC snap-in, focused on Domain Software Installation Policy. Next, drill down to the User Configuration|Software Settings|Software Installation node, where your two applications have been deployed. Then right-click Office 2000 and choose Properties from the shortcut menu. In the Properties dialog box, click the Security tab to show the permissions on the application (see Figure 5.16).

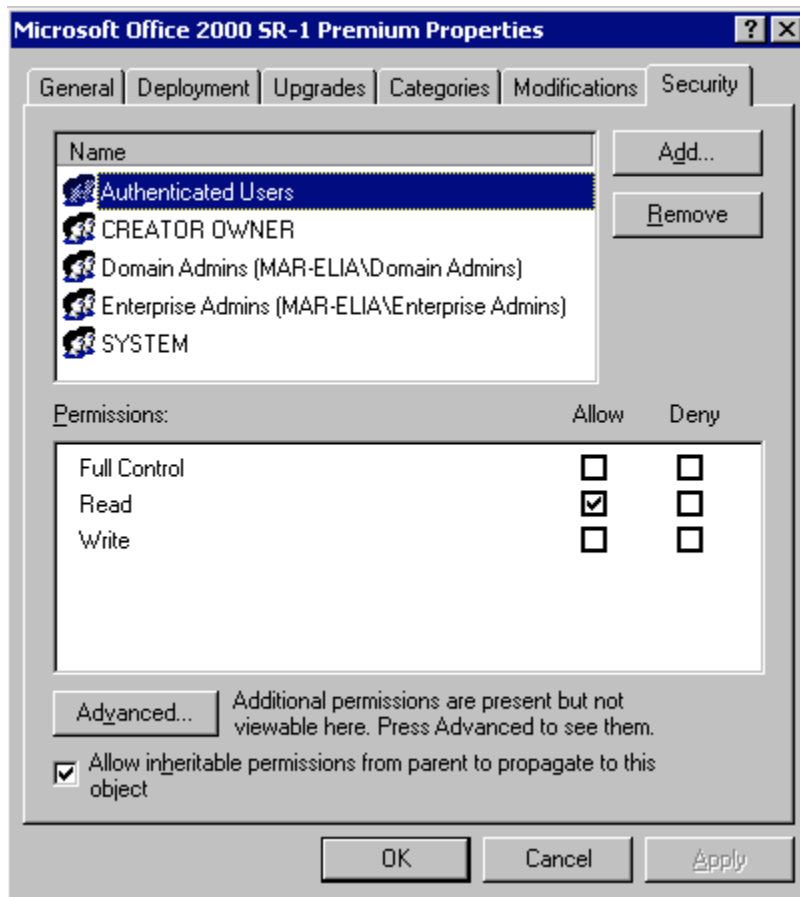


Figure 5.16: Security permissions on applications deployed by GPO-based Software Installation.

The only permission that a user or computer needs to receive the deployed application from the GPO is the Read permission. As you can see in Figure 5.16 above, the Authenticated Users group, by default, has Read permission on all deployed applications. This means that all users who process this GPO will receive these applications. Office 2000 is fine the way it is, but you need to modify the permissions on Win2K Admin Tools to only allow selected administrative groups to receive it.

You edit that application's security permissions just like any other ACL editor operation. In this case, however, first remove the Authenticated Users ACE that grants all users access to this application by clicking Remove in the ACL editor. A message appears, as shown in Figure 5.17.

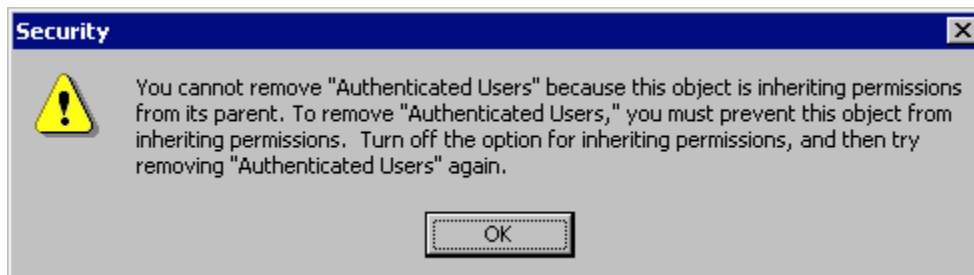


Figure 5.17: The warning message that appears when you try to remove an inherited ACE.

You can't delete the Authenticated Users ACE because it's been inherited from a parent container, so you need to break inheritance between this application and its parent. On the Security page of the Properties dialog box, clear Allow Inheritable Permissions from Parent to Propagate to This Object, then click OK. Another dialog box appears, asking whether you want to copy all inherited ACEs to this object and break inheritance, remove all inherited ACEs from this object, or cancel the operation altogether. Because you don't want to remove all inherited ACEs (just Authenticated Users), click Copy, then Apply.

This breaks the security that the application inherits from its parent container object. However, it may not be apparent what the parent container for an application is. To find out, we have to go back to Chapter 2. In fact, the parent container for an application is a folder in the GPC called Packages. The Packages folder is contained in the Class Store in the GPC (see Figure 5.18).

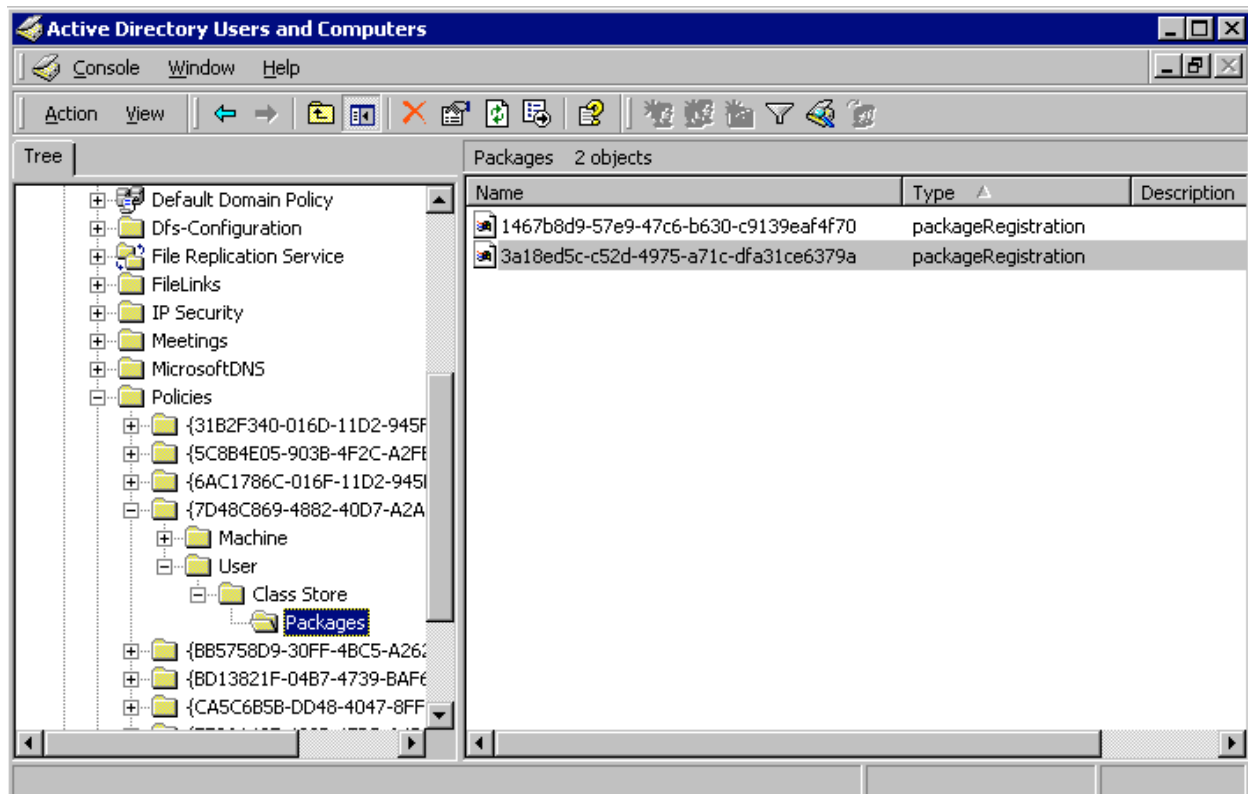


Figure 5.18: The Packages folder in the GPC, from which deployed applications inherit permissions.

As you can see in Figure 5.18, the right-hand (results) pane shows the two applications that are deployed in the GPO, represented as packageRegistration objects in the GPC.

Getting back to our example, once you've broken inheritance between the application and the Packages container, you can remove the Authenticated Users ACE and add your new group permissions. Figure 5.19 shows the resulting ACL for Win2K Admin Tools.

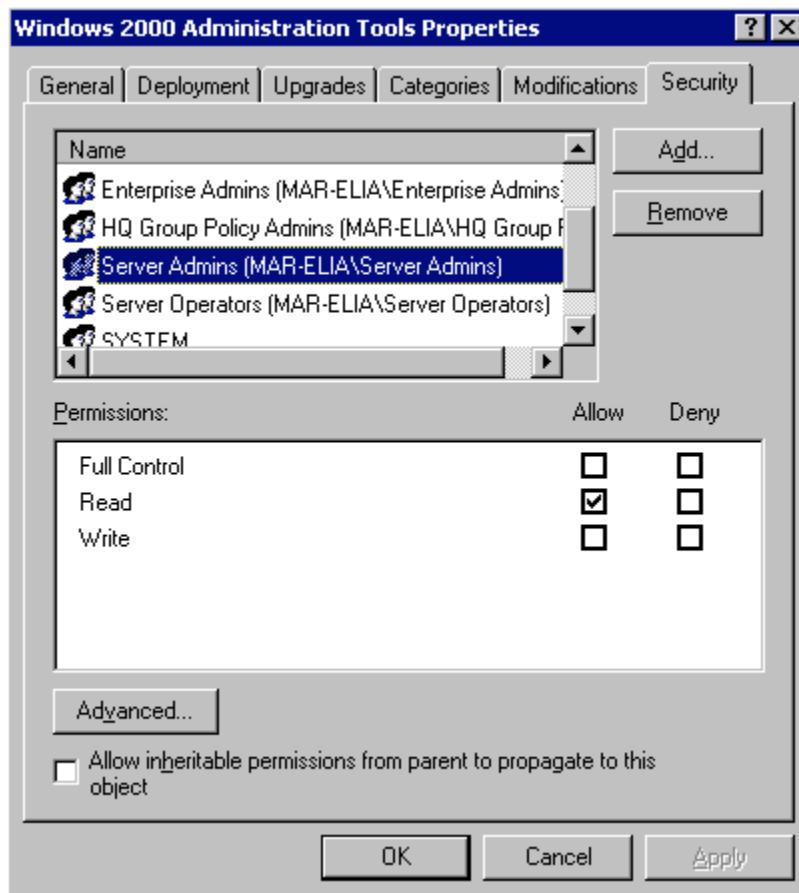


Figure 5.19: The modified permissions on an application object in a GPO.

This figure shows three user groups added —HQ Group Policy Admins, Server Admins, and the built-in Server Operators—and granted each Read permissions to this application. Thus, the next time a member of one of these groups logs on to the domain, Win2K Admin Tools will be advertised to his or her desktop, ready for installation. No other users will receive the application—as you’d expect!

 The example above describes the two applications as per-user deployments. Thus, any security-group filtering you do on these applications needs to be based on *user groups*, rather than *computer groups*, because the applications are processed by users, not computers. If one or both of these applications is deployed per computer (assigned), you’ll need to use computer groups to filter which computers receive which applications.

Application-specific security-group filtering gives you the ability to direct application deployment to specific user or computer groups in a single GPO. Of course, if you’re deploying many applications using GPOs and plan to use security-group filtering on each one, things can get pretty complicated. So, as with other features in Group Policy, balance complexity with capability and make sure that you design an infrastructure that is manageable, well documented, and immune to staff changes.

Using Loopback Processing

When Microsoft was designing Group Policy, several large customers wanted to have a way to override policy settings that would normally apply to users who logged on to a special-purpose computer on the AD network. Special-purpose computers can take a number of forms but include kiosk-type computers, or single-function computers, which present users with a different desktop than they see at their own workstations.

This need also applies to an environment using Windows Terminal Server (WTS) environment. Users log on to Windows terminals using the same credentials they normally use on their own computers, but while they're logged on, you may not want their "home" GPOs to process.

In these cases, Microsoft has provided an interesting workaround—a feature known as *loopback processing*. Loopback processing is enabled on individual computers using a GPO-based Administrative Template policy. When a user logs on to a computer that is using loopback processing, the computer overrides the user account's own "home" GPOs and temporarily applies any GPOs that it processes instead.

To enable loopback processing, you need to open a Group Policy snap-in focused on a GPO and drill into Computer Configuration|Administrative Templates|System|Group Policy. Figure 5.20 shows where the loopback policy setting is located in a GPO.

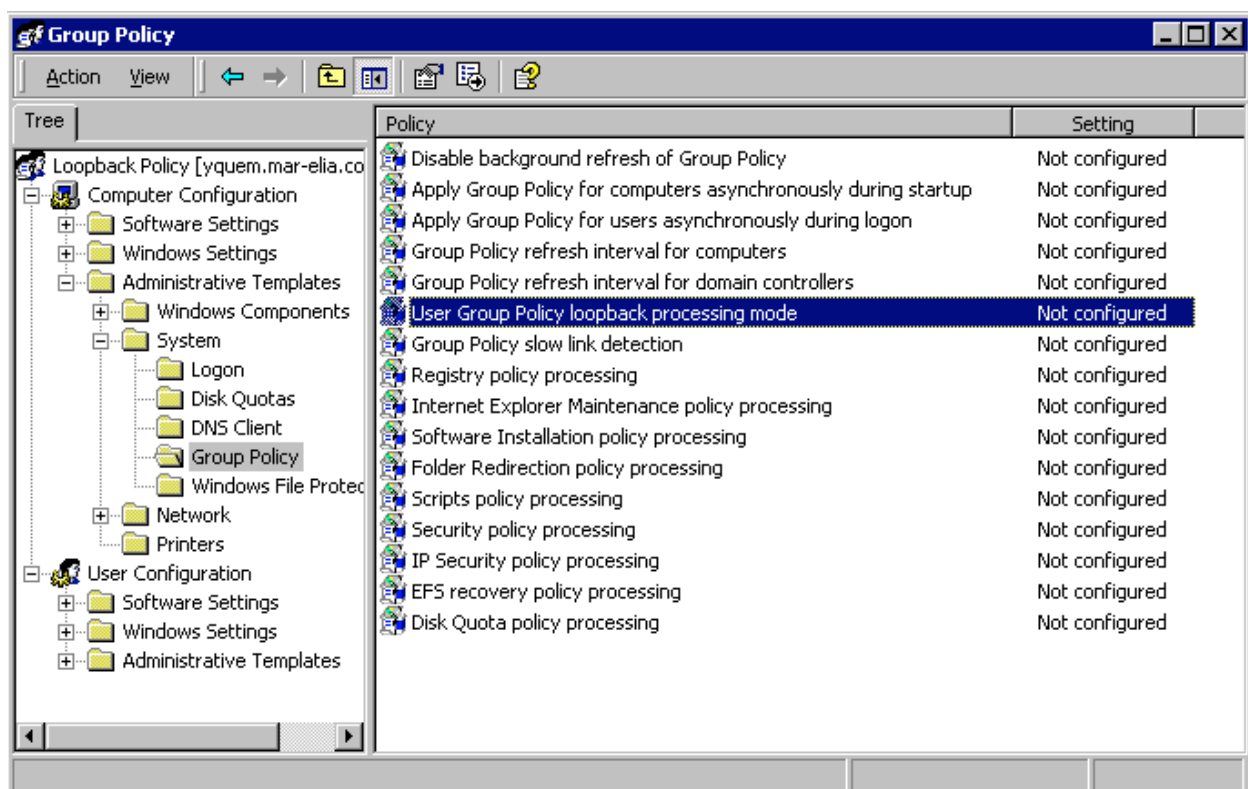


Figure 5.20: Loopback policy settings in Administrative Template policy.

Loopback can work in two modes—Merge and Replace. Each mode determines how user-specific policy is merged into a user's existing user-specific GPO-based settings. In Merge

mode, when a user logs on to a computer that has loopback processing enabled, any user-specific GPO policy that the computer is processing will be merged with any policy that the user brings from GPOs that he or she normally processes. Figure 5.21 attempts to illustrate how this works.

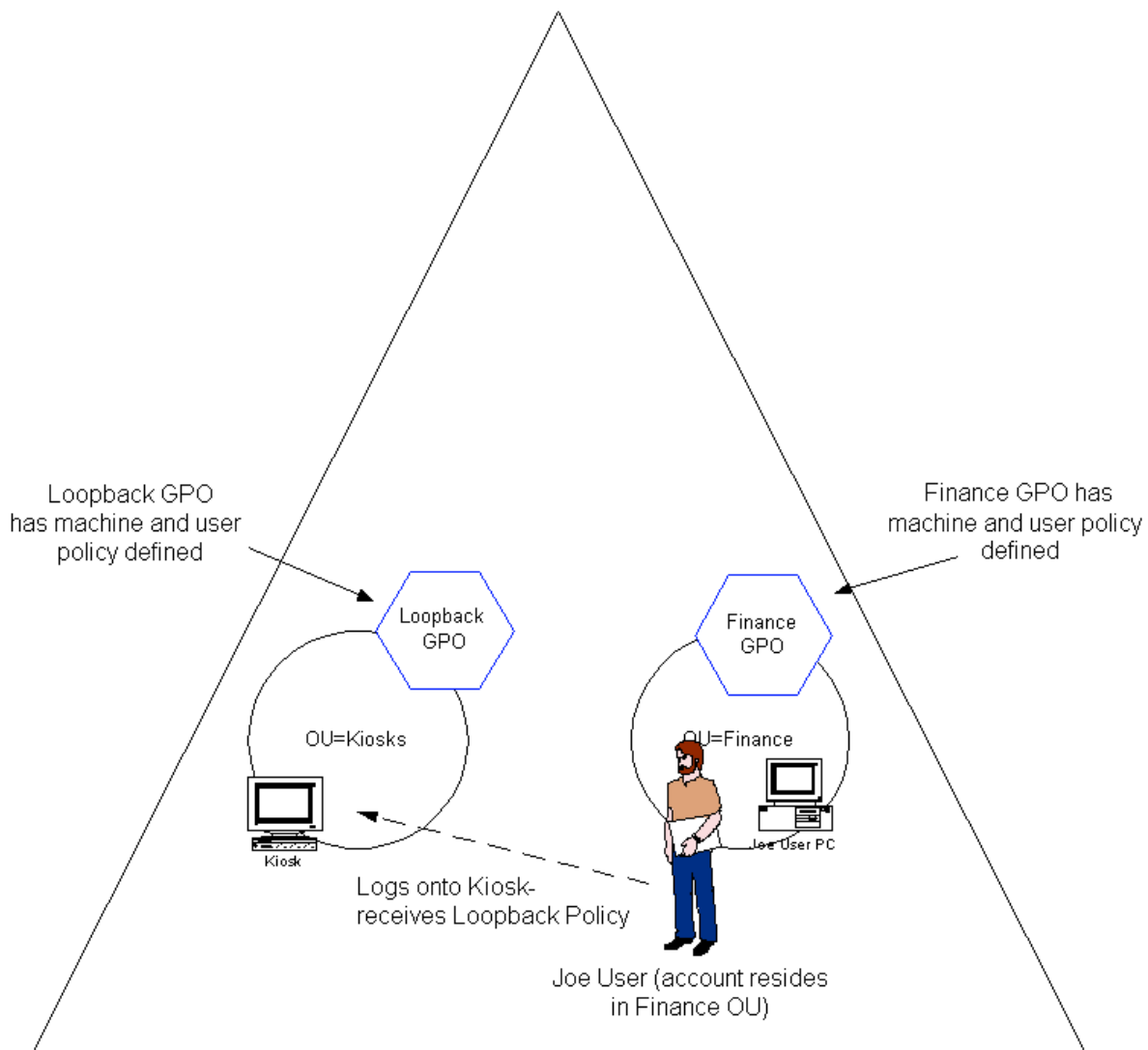


Figure 5.21: How GPO loopback processing works.

In the figure above, Joe User has an account in the Finance OU, and he normally logs on to a computer in that OU as well. A Finance GPO provides Joe's computer- and user-specific policy. User-specific policy may include Folder Redirection, Administrative Template lockdown, Internet Explorer (IE) Maintenance policy, and so on. When Joe User logs on to the computer labeled Kiosk in the Kiosks OU, the user-specific policy that he typically receives in his own OU can be overridden using loopback processing.

If you use loopback processing in Merge mode, any user-specific policy settings that have been defined on the Loopback GPO will be merged with Joe's "home" policy settings—those that were applied from the Finance GPO. For example, perhaps Joe normally receives a Folder

Redirection policy from the Finance GPO that redirects his My Documents folder to a server share but leaves other folders (like Start Menu, Desktop, and so on) where they are by default.

Now let's say that in the Loopback GPO, you create a Folder Redirection policy that redirects all users' Desktop folders to a server share. Using Merge mode, the result is that Joe's My Documents and Desktop folders are redirected to server shares. If any of these user policy settings conflict during the merge process, the policy defined on the Loopback GPO will override what was defined on the Finance GPO.

If you use loopback processing in Replace mode, any user policy that Joe User receives from the Finance GPO will be completely replaced by what is defined on the Loopback GPO (see Figure 5.21). Joe's My Documents folder will no longer be redirected to the server share; his Desktop folder will be redirected instead.

To enable loopback processing, make sure you're in the Group Policy snap-in as shown in Figure 5.20, then open User Group Policy Loopback Processing Mode. In the Properties dialog box, click Enabled, then specify a mode (see Figure 5.22).

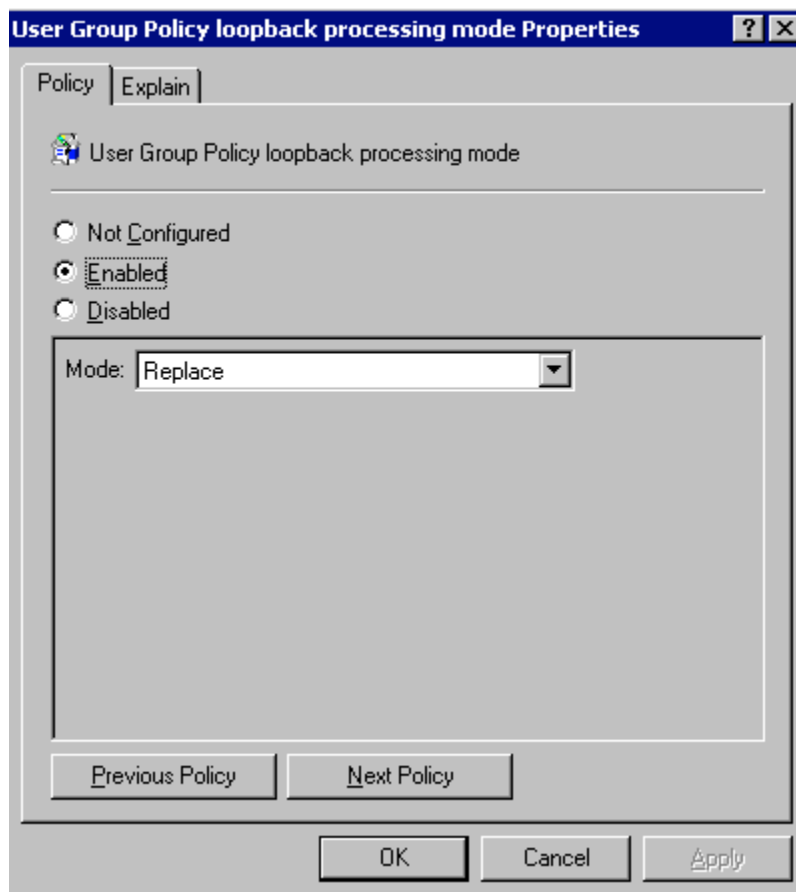


Figure 5.22: Setting loopback processing with a GPO.

You can see how loopback processing can be ideal in a Windows terminal environment. You can lock down certain aspects of a user's desktop when he or she logs on to a terminal but let the user

“roam free” when he or she is sitting at their own computer. To use this approach, simply enable loopback processing using one of the two modes I described above on a GPO that is processed by your terminal server(s). Then, either on that same GPO or on different ones, define a set of user-specific policy settings that will take effect when any user logs on to a terminal.

I recommend that you start using loopback processing in Replace mode rather than Merge mode. As you can imagine, calculating the Resultant Set of Policy (RSOP) for a user who is receiving the results of multiple merged GPO settings can be a nightmare unless you have a really good lock on what policies you’ve set in your environment. Using Replace mode, you always know that the only policies a user is getting are the ones defined on the GPOs that are processed by the computer with loopback processing enabled.

Summary

In this chapter, I described ways to work around some challenging issues in your GPO infrastructure. Delegation, security-group filtering, and loopback processing are all tools that, once you master them, can make using GPOs more worthwhile. When you use delegation, the key is to understand the differences among delegating creation, editing, and deletion of GPOs, and linking and un-linking them, and what permissions enable each.

I also discussed the GPCO security group and some important differences between the DoC wizard and setting ACLs directly. With respect to security-group filtering of GPO processing, the key is to keep it simple and balance security-group filtering with proper GPO linking. Finally, loopback processing gives you a way of overriding users’ native user policy settings when they log on to specialized computers in your AD infrastructure. In the next and final chapter, I’ll look at tools and techniques for managing and troubleshooting your GPO infrastructure.