Windows 2000 Group Policies

Using Group Policies, administrators can establish and maintain a wide range of user and computer configuration settings. Management is simplified because policies only need to be stipulated once.

By Dave Cook
Technical Consultant And Journalist

Windows 2000 provides plenty of administration tools, though few are quite as powerful as the Group Policy snap-in. Group Policies allow administrators to create specific desktop configurations for a particular group of users or computers. The users' environment needs to be stipulated only once, with the operating system enforcing the policy thereafter. The result is an easier, more streamlined administration, and a lower total cost of ownership. Typically, Group Policy settings might define different Start menu options, the types of programs available to users, and the programs that appear on the user's desktop. Administrators could use Group Policy to set a local disk quota limit, redirect a user's My Documents folder over the network, and define logon and logoff scripts, as well as control numerous security settings.

Group Policy Objects

Group Policy settings are contained in Group Policy objects, which are similar in character to Microsoft .DOC files or .TXT files with Notepad. The analogy is not perfect, however, as changes to Group Policy objects take place during the actual edit, rather than when an explicit Save is executed. But before administrators can define Group Policy settings, they must first create one or more Group Policy object(s). The objects are in turn associated with selected Active Directory objects: domains, sites, and organisational units.

Settings are stored inside Group Policy objects in two main areas: the user configuration and the computer configuration. User configuration settings typically consist of policies that affect users, such as the behaviour of the Start menu, desktop, and display areas. Computer configuration settings are made up of policies that control the registry, file system, and other computer-based settings. Computer configuration settings are applied when Windows 2000 initialises, whilst user configuration settings are applied during the logon process.

Hierarchical Processing

By default, Group Policy is processed hierarchically, beginning with policy stored in the highest-level container in Active Directory. This means that the settings applied to users and computers are inherited from site to domain, and then on to the organisational unit level. The order in which Group Policy objects are applied determines which settings a user or a computer actually receives. Thus, when two policy settings are in conflict with each other, the settings defined in the policy closest to a user or computer will overrule the other.

But what if administrators have no wish to see that happen? No problem; the default behaviour for inherited Group Policy can be changed in several ways. For example, administrators could use permissions to alter the behaviour of Policy objects. They could also block Group Policy containers at site or domain level, or they could override Group Policy defined in the lower-level containers of organisational units. Note, however, that when the block Group Policy settings and the override Group Policy settings combine, the override Group Policy setting will always apply. This allows administrators who have control of the high-level containers in Active Directory to enforce the Group Policies found in lower-level containers.

A clear if rather unpretentious example of how policy hierarchy is used to set precedence for conflicting policy entries can be found in the following rule: if local policy is set to use a purple desktop background, but there is a site policy that dictates a yellow background, a domain policy that sets a blue background and an organisational unit set for green, the result of all those conflicting entries is a green desktop.

NT4 And Win 2000

The ability to control the user work environment and enforce system configuration settings is not new, of course, Microsoft having introduced the System Policy Editor to Windows NT 4.0. However, anyone who has used this editor will most likely be all too aware of its limitations. Basically, System Policy settings are registry settings that define the behaviour of various components on the desktop.

Unfortunately, these settings are not secure, and any user who knows how to use the registry editor, REGEDIT.EXE, can change them with ease. Another drawback to System Policies is that the settings persist until each is reversed (or the user edits the registry). In Windows 2000, however, Group Policy settings cannot persist past their useful life. This is because the operating system removes them when a Group Policy object no longer applies. Moreover, only an administrator can change these settings which, by the way, are held in the following secure registry locations: \Software\Policies and \Software\-Microsoft\Windows\-CurrentVersion\Policies.

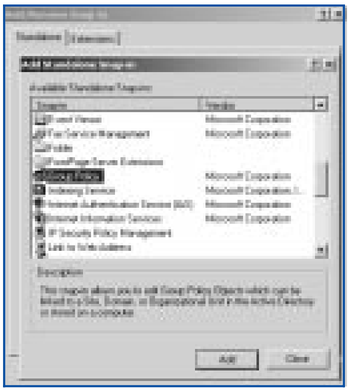

Figure 1 - The Add Standalone Snap-In dialog box.

The files used to support Group Policies are not actually retained in the Active Directory database. Instead, they are stored as discrete policy folders under \WINNT\SYSVOL\SYSVOL\<domain_name>. Windows 2000 Server creates the SYSVOL (system volume) folder automatically upon installation of a domain controller, or when a server is promoted to be a domain controller. The topmost folder in this structure is named after the globally unique identifier (GUID) of the Group Policy object, while the remaining folders store the Policy object's computer and user settings. Thus, when a Group Policy object is created in the Active Directory, several folders are added to the second SYSVOL folder of the domain controller. This is a shared folder that allows the information to be replicated to all domain controllers belonging to the same domain. Client computers and users access the second SYSVOL folder to find policies.

When Windows 2000 clients log on to the domain, they query the Active Directory for Group Policy container objects. If any of these objects is applicable to the client, the associated policy folder contents are then downloaded from SYSVOL. Should the policy affect the registry - and most do in one way or another - then the policy entries will overlay the existing registry entries. However, contrary to the System Policy of Windows NT, if the policy is removed then the underlying registry entries will return to the way they were before the entries were applied.

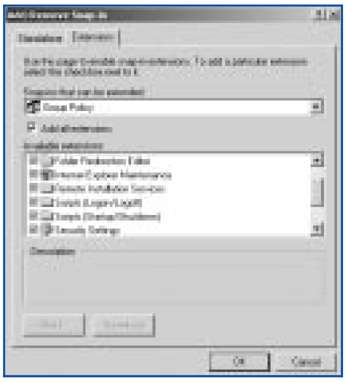

Figure 2 - Use the Extensions tab to list all available selections.

Snap-Ins

Administrators can create non-local Group Policy objects by using the Group Policy snap-in, either as an extension to Active Directory snap-ins, or as a standalone MMC console. The most common route to a Group Policy snap-in is from Active Directory Users And Computers. From here, administrators can link Group Policy objects to domains or organisational units. By accessing Group Policy through Active Directory Sites And Services, they can also link Group Policy objects to sites. From these two Active Directory consoles, Group Policy is accessible by means of a context menu. To access this menu, administrators would right-click the site, domain or organisational unit, point to Properties, and click the Group Policy tab.

Note that Group Policy objects highest in the links list have precedence over the lower objects. Administrators wishing to change the domain policy settings, for example, could select the Default Domain Policy, click Edit, and then, from the Group Policy window, check the list of preferences under each of the Settings objects. Selecting the Options button, meanwhile, allows administrators to disable the object. They can also prevent other Policy objects from overriding policy settings in the container.

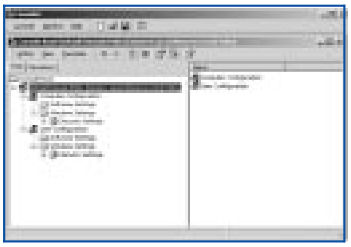

Figure 3 - The view from the MMC after adding Security Settings.

Alternatively, administrators could select Properties, and then click the Security tab for a quick run-down of the users and groups to whom permissions have been assigned for the object. Note also that this is where domain administrators select groups of users to grant them Read / Write access to specific Group Policy objects, after which the users in turn can take control over the objects.

Somewhat surprisingly, and despite a long list of standalone consoles readily accessible from the Administrative Tools section in Windows 2000, Microsoft has chosen not to make the Group Policy Editor available as a standalone console. However, opening an empty Management Console (MMC) and manually adding the Group Policy snap-in can rectify this exclusion quite easily. After this the Editor is made accessible from the Start button by placing it in the same folder that contains the rest of the Administrative Tools.

Adding The Group Policy Editor

Adding the Group Policy Editor to the Administrative Tools folder can be achieved like this: click Start, Run, and in the Run dialog box type MMC. From the empty console's menu bar, select Console, Add / Remove Snap-In, and then, from the Standalone tab, click Add. This opens the Add Standalone Snap-in dialog box (Figure 1) and displays a variety of ready-made policy elements, any of which can be added to a console. Select Group Policy in the list of available snap-ins, and click Add to launch the Select Group Policy wizard. The wizard initially offers administrators the chance to create a local computer Group Policy object.

Note that this is not a Group Policy as such; instead, it looks directly at the local policy database in \WINNT\System32\GroupPolicy. Now click Finish to accept the default Local Computer Policy object, click Close, and then click OK to install the snap-in into the console. On the Console 1 menu bar, choose Console, Save As, and in the dialog box navigate to the Administrative Tools folder and type Group Policy in the Filename text box. Finally, click Save and close the console.

Policy Elements

Broadly speaking, there are five main policy elements designed to help administrators gain the most benefit from Group Policy. Administrative templates allow administrators to configure registry-based policy; the folder redirection element allows specified folders to be distributed on the network; a scripts policy specifies scripts that control user logon and logoff policy, as well as regulating computer startup and shutdown procedures; the software installation policy allows administrators to selectively advertise and publish applications; and finally the security settings policy is used to configure security for users, computers, and domains on the network.

The settings defined in Group Policy objects are quite comprehensive. They include objects in Active Directory, account policies, local policies, the registry, IP Security, the file system, restricted groups, system services, and others. For the purpose of this next exercise we shall add the Default Domain Policy with security settings to the MMC, noting that any adjustments to the Security Settings extension can be made later, as and when necessary.

Further Information

Microsoft hardware compatibility list
www.microsoft.com/hcl

Development and testing site
www.microsoft.com/hwdev

Latest application compatibility information
www.microsoft.com/windows2000/compatible

Microsoft's planning and deployment guide
www.microsoft.com/windows2000/library/planning

Windows 2000 Resource Kit selection
www.microsoft.com/windows2000/library/resources/reskit[ends]

Administrative Templates

Entries that define Computer and User Group policies are taken from a collection of administrative templates. These templates are a special type of ASCII file. They contain the same format as their classic NT System Policy counterparts, but with additional features and some new functionality.

Administrative templates determine and control which registry settings can be modified with Group Policy Editor snap-ins. As such, they consist of a group of categories (and occasionally subcategories) that control how the Group Policy Editor presents certain options to the administrator in the MMC. When selected, each template automatically identifies any registry locations where the system needs to make changes, specifying any additional options or restrictions that may also be required.

Some templates, such as WINNT.ADM and WINDOWS.ADM, are provided for backward compatibility with Windows NT and Windows 9x, while the INETRES.ADM template typically controls policies affecting Internet Explorer. Alternatively, the SYSTEM.ADM template includes a comprehensive set of system restrictions, helping adminis*trators take control of a wide variety of Desktop, Control Panel, Network and System settings.

Adding The Default Domain Policy

To add the Default Domain Policy with security settings to the MMC, return to the Add Standalone Snap-In dialog box. Click Group Policy in the list of available snap-ins, and click Add to launch the Select Group Policy wizard. Now click the Browse button to find a different Group Policy object (note that browsing through the other tabs in the dialog box will show more detail). Select the Default Domain Policy from the Browse For Group Policy Object window, and click OK to return to the Select Group Policy window. The Default Domain Policy will now appear under the Group Policy Object. Click Finish to add the policy and return to the Add Standalone Snap-In window, then click Close to return to the Add / Remove Snap-In window.

Click on the Extensions tab (see Figure 2) and uncheck the Add All Extensions box. Note that this option must be unchecked before administrators can edit the list of available selections. Next, go down the list of available extensions, ensuring that only the Security Settings option remains checked. On completion, click OK to save the change and return to the console (see Figure 3). From the main console menu, choose Console and select Options. When the Options window opens, select the Console tab and give the icon a name, such as Domain Security Policy Editor.

In the Console mode box, verify that Always Open Console Files In Author Mode is not selected. Administrators should normally try to avoid Author Mode since this mode makes it all too easy for users to add and delete snap-ins from the console. Usually it is better to select User Mode - Full Access, Single Window. Although this option gives users the ability to change the window's focus, it prevents them from loading additional snap-ins and extensions. Note that there are three other options available at the bottom of the Options window: Enable Context Menus On Taskpads In This Console (checked); Do Not Save Changes To This Console (unchecked); Allow The User To Customize Views (checked). Generally, these three options are best left the way they are by default.

Click OK to save the changes. On returning to the console, close the console and click Yes to save the settings. Give the file a short but descriptive name, such as DOMSECED.MSC, and save it to the Documents And Settings \ All Users profile. This makes it visible in the Start menu to anyone logging on to the computer. If this is not required, saving it to Administrative Tools will suffice. As a final check, open the saved console and ensure that the main console menu is not visible. This shows that the console has not opened in Author mode.

In due course, there may come a time when changes to the console are deemed necessary. In which case the file can be opened in Author mode via the following method: click Start, Run, and in the Run dialog box, type MMC DOMSECED.MSC /a, where DOMSECED.MSC is the filename. Be aware, however, that the /a switch only works with administrator privileges.

Conclusion

Group Policy objects should be designed to cater for the unique needs of the organisation. In some companies, a single Group Policy object will suffice. In medium to large companies a number of Group Policy objects may be required. Often, it will make sense to build a custom Group Policy console and load only those snap-in extensions deemed necessary.

Sometimes it will be necessary for administrators to farm out security policy to one group, for example, and a software distribution policy to another group. However, delegating administrative privileges is a sensitive and dangerous business. In the wrong hands, such privileges can devastate even the most stable and secure systems. Always ensure, therefore, that only the organisation's top administrators have the ability to hand over such powers to other users.

> *"Group Policy objects should be designed to cater for the unique needs of the organisation. In some companies, a single Group Policy object will suffice."*

PCSA