C H A P T E R 1 8

Authorization and Access Control

Thank you for reviewing the Whistler Server Resource Kit documentation. Post your feedback at the **microsoft.betanews** account under the **microsoft.beta.whistler.documentation** newsgroup using the following steps:

1. Create a new newsgroup message.
2. Use the title of the chapter as the subject line of your message.
3. Put your comments in the message, clearly identifying page number and changes to the text.

You can also send the chapter with your comments to docbeta@microsoft.com.

The Microsoft Whistler operating system includes a number of features that you can use to protect selected files, applications, and other resources from unauthorized use. These features, which include Access Control Lists, security groups, and Group Policy, along with the tools that allow you to configure and manage these features, provide a powerful, yet flexible access control infrastructure for your network. Understanding what these features are, why they are necessary, and how they function will help you to manage privileges and permissions on network and local resources more effectively.

In This Chapter

Related Information in the Resource Kits

- For more information about the authentication process and how security contexts are created, see "Logon and Authentication" in this book.
- For more information about authorization in Active Directory environments, see "Authorization and Access Control" in the *Distributed Services Guide* of the *Microsoft Whistler Server Resource Kit*.

Overview of Access Control

Every user and computer has a specific role and purpose in an organization. In order to accomplish their goals, each user and computer must be able to access certain resources and perform specific tasks. However, allowing users and computers unlimited access to system and network resources and functionality can compromise an organization's security and stability. Whistler's access control infrastructure functions to balance the resource access and system security needs of an organization.

For example, Alice works in Accounting and needs to be able to view — but not create or modify — certain Personnel Department files that are off limits to other users in the organization. The Personnel department, which controls these files, has used access control to define which users can have Read-only access to Personnel files, which users can have Write and Modify access, and which users have no access to the Personnel share. Alice has been given Read-only access to the Personnel files. At the same time, IT has determined that prohibiting users such as Alice from making significant changes to their systems can reduce costs and improve security and supportability. IT has made Alice and other users members of the Users group, thus limiting their ability to install applications and reconfigure their operating system environments. In this way, Alice has the access to resources that she needs, and the security of the organization is maintained.

Key Terms

In order to understand the basic principles of access control, it is important to understand how the following key terms are defined in the context of the access control model for Windows 2000 and Whistler.

Security principal. A user, group, computer, or service. Security principals have accounts. Local accounts are managed by the Security Accounts Manager (SAM) on the computer. If

the account is in a native Windows 2000 or Whistler domain, it is managed by Active Directory. If the account is in a Windows NT 4.0 domain, it is managed by a SAM database on the domain controller.

Security identifier (SID). A value that uniquely identifies a user, group, service, or computer account within an enterprise. Every account is issued a SID when it is created. Access control mechanisms in Windows 2000 and Whistler identify security principals by SID rather than by name.

Security context. Information that describes a particular security principal's identity and capabilities on a computer. In Windows 2000 and Whistler, all users in an organization exist in a specific security context that is reestablished every time they log on. All activities, such as installing or running applications, take place in this security context. The security subsystem uses the security context to determine what a process and its threads of execution can do to objects on the computer, and who will be held accountable for what they have done.

*Access token***.** A data structure containing the SID for a security principal, SIDs for the groups that the security principal belongs to, and a list of the security principal's rights on the local computer. An access token is created for every security principal that logs on locally at the computer's keyboard or remotely through a network connection. The access token provides a security context for the security principal's actions on the computer. It also provides a security context for any application threads that act on the security principal's behalf.

Object. Any resource that can be manipulated by a program or process. Objects include resources that you can see through the user interface, such as files, folders, printers, registry keys, Active Directory objects, and the Windows desktop. They also include resources that you cannot see, such as sessions, processes, threads, and access tokens. An object can function as a logical container for other objects.

Inheritance. A mechanism for propagating access control information down through a tree of objects. In Windows NT, an object (such as a file) inherits access control information from its parent object (such as a folder) only when the object is first created. In Windows 2000 and Whistler, objects inherit access control information not only when they are created, but also when the parent object's access control list changes.

Owner. The only security principal who has an inherent right to allow or deny permission to access an object. An object's owner can give another security principal permission to take ownership. By default, the built-in Administrators group on a computer is assigned a user right that allows this group to take ownership of all objects on the computer.

Security groups. Groups that can be used to organize users and domain objects, thus simplifying administration. Security groups allow you to assign the same security permissions to a large numbers of users, such as employees in a single department or in a single location, ensuring that security permissions are consistent across all members of a group.

Security descriptor. A data structure containing the security information associated with a securable object. A security descriptor identifies an object's owner by SID. If permissions are configured for the object, its security descriptor contains a discretionary access control list (DACL) with SIDs for the users and groups that are allowed or denied access. If auditing is configured for the object, its security descriptor also contains a system access control list (SACL) that controls how the security subsystem audits attempts to access the object.

Access control list (ACL). An ordered list of access control entries (ACEs) that define the protections that apply to an object and its properties. Each ACE identifies a security principal and specifies a set of access rights allowed, denied, or audited for that security principal.

Security settings. Security configuration settings that can be applied to individual computers. These settings can be configured locally on the computer using the Local Security Policy administration tool, the Security Configuration and Analysis snap-in to the Microsoft Management Console (MMC), or, if the computer is a member of an Active Directory domain, through the Security Settings extension to Group Policy.

Key Concepts

The security systems in Windows 2000 and Whistler are based on technologies originally developed for Windows NT. The access control models in Windows NT, Windows 2000, and Whistler share the same key concepts and characteristics, which are described in the following sections.

Discretionary access to securable objects The user who owns an object has ultimate control over who has permission to use it and in what way. An object's owner can

give permission for different kinds of access to particular users or groups of users. For example, the owner of a file object can give Read and Write permission to all members of one group while denying Write access to members of another group. In Windows 2000 and Whistler, owners can Allow or Deny other users access to individual properties of certain types of objects as well as to the entire object. The properties that can be delegated include the ability to Allow or Deny other users access to the object.

Inheritance of permissions You can control permissions for new objects created in a container object by setting inheritable permissions on the container. The permissions that you set on a container are inherited by existing objects in the container, as well as by newly created objects. For example, the permissions that are set on an NTFS folder are inherited by new subfolders and files created within the folder.

Auditing of system events You can use the auditing feature to detect attempts to circumvent protections on resources or to create an audit trail of administrative actions on the system. For example, you can audit failed attempts to open a file. You can also set security policy so that failed logon attempts are recorded in the security event log. If another administrator changes the auditing policy so that failed logon attempts are no longer audited, the log can record this event as well. In Windows 2000 and Whistler, you can use Group Policy to centrally control who is allowed to manage security logs on computers joined to a domain.

Rights and Permissions

Access control involves the configuration of *rights* and *permissions*, which apply to both the objects on the computer or network and the potential users (including individuals, computers, and services) of those objects.

A right is authorization to perform an operation. In Whistler, one right is inherent — the right to allow or deny access to resources that you own. All other rights must be granted, which means that they can also be withdrawn. From an administrator's point of view, there are two types of rights: permissions and user rights.

A permission is authorization to perform an operation on a specific object, such as a file. Permissions are granted by owners. If you own an object, you can grant any user or security group permission to do whatever you are authorized to do with it.

When permission to perform an operation is not explicitly granted, it is implicitly denied. For example, if Alice allows the Marketing group, and only the Marketing group, permission

to read her file, users who are not members of the Marketing group are implicitly denied access. The operating system will not allow users who are not members of the Marketing group to read the file.

Permissions can also be explicitly denied. For example, Alice might not want Bob to be able to read her file, even though he is a member of the Marketing group. She can exclude Bob by explicitly denying him permission to read the file. In fact, this is exactly how explicit denials are best used — to exclude a subset (such as Bob) from a larger group (such as Marketing) that has been given permission to do something.

Each permission that an object's owner grants to a particular user or group is stored as an ACE in a DACL that is part of the object's security descriptor.

User-Based Authorization

Every application that a user starts runs in his or her security context, not in its own security context.

When a user logs on, an access token is created. The access token contains key security-related information, including the user's SID, the SIDs of the groups to which the user belongs, and other information about the user's security context. This access token is then attached to every process that the user runs during that logon session.

An application runs as a process with threads of execution. When an application performs an operation on a user's behalf, one of the threads performs the operation. For example, when Alice opens a Word document, Microsoft Word, and not Alice, actually opens the file. More precisely, one of the threads of execution performs the operation.

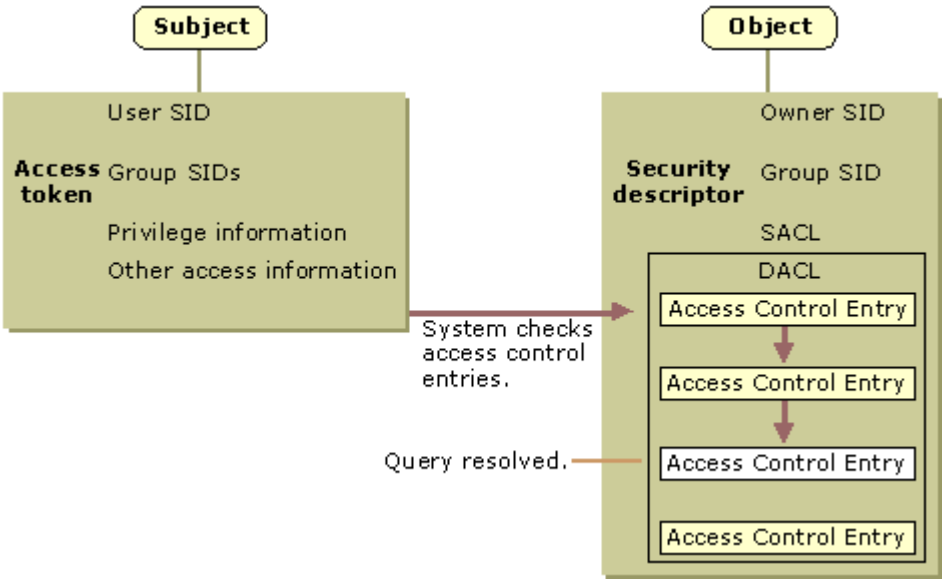
In order for a thread to gain access to an object such as a file, it must identify itself to the operating system's security subsystem. Threads and applications do not have a security identity, so they must borrow one from a security principal, such as Alice. When Alice starts an application, it runs as a process within her logon session. When one of the application's threads needs to open a file, the thread identifies itself as Alice's agent by presenting her access token. Alice is therefore ultimately responsible for anything that the thread does to the file or system on her behalf.

Before allowing the thread of execution to proceed, the operating system performs an access check to determine whether the security principal associated with the thread has

the degree of access that the thread has requested. This access check involves the following steps:

1. The security subsystem checks the file object's DACL, looking for ACEs that apply to the user and group SIDs referenced in the thread's access token.
2. If a DACL does not exist, access is granted. If a DACL exists, but is null, access is denied. Otherwise, the security subsystem steps through the DACL until it finds an ACE that either allows or denies access to the user or one of the user's groups.
   - If access is allowed, the file is opened.
   - If access is denied, the file remains closed and an **Access Denied** message is generated.
3. If the security subsystem comes to the end of the DACL and the thread's desired access is still not explicitly allowed or denied, the security subsystem denies access to the object. Figure 18.1 illustrates this process.

Figure 18.1 Validating a request for access



In the case of the Personnel files, Alice's administrators set a DACL on the folders and files that she needs to work with to explicitly define the extent (Read) or limits (not Create or Write) of access that she as an individual or member of a security group has to those files.

Every computer and service on the network also has a security context that governs the resources that it is permitted to access and the actions that it is permitted to take. In Whistler, applications can be designed to run in restricted security contexts, giving them fewer privileges and more limited access than their users have.

Security Descriptors

Access control information is first written to an object's security descriptor when the object is created. Then, when a user tries to do perform an action with the object, the operating system examines the object's security descriptor to determine whether the user is allowed to do what the user wants to do.

The information that is included in a security descriptor depends on the type of object in question and how it was created. In general, security descriptors can include the following information:

- Which user owns the object
- Which users and groups are allowed or denied access to the object
- Which users' and groups' access to the object must be audited

This information can later be modified. In both cases, the information that goes into a security descriptor can come from one of the following sources:

- The subject
- The parent object
- The object manager

When a subject creates a new object, it can assign the object a security descriptor. If the subject does not assign a security descriptor, the operating system uses access control information inherited from the parent object to create one. If no information is available to inherit, the operating system uses default access control information provided by the object manager for the particular type of object that the subject wants to create.

After an object is created, the object's owner or another user who has the permission to change permissions can change information in the object's security descriptor. The owner can assign the permission to change permissions to other users. Changes can also come

from the parent object when that object's owner modifies its security descriptor. This process is called inheritance. Every time the security descriptor on a container object is changed, the object manager propagates any changes marked as inheritable to all objects in the container, as long as those objects are not protected. For more information about managing inheritance, see "Inheritance and DACLs" and "Inheritance and SACLs" later in this chapter.

Planning for Effective Access Control

Managing security groups, ACLs, and security settings requires a great deal of initial planning. Developing an access control plan can help to prevent basic security problems, such as inadequately protected resources, users granted greater rights and permissions than they need to do their jobs, or ad hoc security configurations that are not based on a well-thought-out, manageable security plan. Ad hoc security management might provide adequate protection for small organizations, but will quickly break down as the organization grows.

Although Whistler incorporates highly advanced security technologies, effective access control must combine the proper use of Whistler technologies with good planning. Security technology is only as good as the methods used to employ and manage it.

> **Tip**
> To improve the security of your network, provide each user, computer, and service with the least amount of privileges needed to perform their tasks. Whistler includes improved features — including well-defined default security groups, Restricted Software settings, and the Secondary Logon Service (SLS) — to make this possible. For information about SLS, see "Authentication" in this book. For information about Restricted Software settings, see "Software Restriction Policies" later in this chapter.

Consider developing an access control plan that describes how you will use Whistler features to establish a secure, usable environment. A typical access control plan might include the following sections:

- **Security goals**. Describe the resources and processes that you are protecting.
- **Security risks**. Enumerate the types of security hazards that affect your enterprise, including what poses the threats and how significant these threats are.

- **Security strategies.** Describe the general security strategies necessary to meet the threats and mitigate the risks.
- **Security group descriptions.** Define the security restrictions or permissions that might apply to different groups of users and resources, and then create security groups to help you implement these sets of permissions and restrictions.
- **Security policy.** If you add your Whistler Professional clients to a Windows 2000 or Whistler Active Directory environment, you can use the Security Settings extension to Group Policy to define and enforce your security strategy on any number of computers.
- **Information security strategies.** Define how you plan to implement information security solutions, such as an encrypting file system (EFS), and access authorization using permissions. For more information about EFS, see "Encrypting File System" in this book.
- **Administrative policies.** Document policies for delegation of administrative tasks and monitoring of audit logs to detect suspicious activity.

Your access control plan can contain additional sections, but these are suggested as a starting point. If possible, test and revise your access control plans using a test laboratory that closely resembles your organization's computing environment. Also, conduct pilot deployments to further test and refine your access control plans.

> **Tip**
> For more information about planning for and deploying security technologies, see *Designing Directory and Security Services* in the *Microsoft Whistler Deployment Resource Kit*.

User Accounts and Security Groups

Creating and deleting user accounts and defining and using security groups are major security tasks. Defining the security restrictions or permissions that might apply to different groups of users and resources in your network will help to simplify the implementation and management of the permissions and restrictions in your organization. For example, you can

create a Printer Operators group and give it precisely delineated administrative control over a finite group of printers.

In order for you to effectively manage security groups in your organization, you need to be familiar with the relationship between accounts, security groups, and special identities. It is also important for you to become familiar with the techniques and tools available for managing group membership.

> **Note**
> Security groups are a built-in feature of Whistler. No special installation or prerequisite is required for you to utilize security groups.

User Account Creation

Every user has an *account,* which contains unique credentials that allow the user to access resources on a local computer or domain. Accounts can be local to a computer or domain based. If the account is specific to a local computer, the user will not be able to access network-based resources, unless certain resources have been configured to allow Anonymous access. If the account is domain based, the user will be able to access network resources by using a local computer. However, his or her permissions as a user of network resources might be quite different than his or her rights on the local computer. For more information about how accounts are authenticated, see "Logon and Authentication" in this book.

Two user accounts — Administrator and Guest — are created automatically when Whistler Professional is installed. The Administrator account can be used to initially log on and configure the computer. For example, the administrator can install software, configure printers, join the computer to a domain, and so on. After the computer has been configured, it is only necessary to log on as the administrator when you must perform administrative tasks.

> **Tip**
> It is best if the Administrator account has a strong password. You can also rename the Administrator account to make it more difficult for potential hackers to gain access to your system.

The Guest account can be used to allow different users to log on and access local resources without having to create an account for each user. The Guest account can also be enabled

to simplify file and printer sharing with other Windows computers that are configured in a workgroup environment.

> **Note**
> The built-in Guest account is disabled by default.

User accounts are not created automatically when the operating system is installed. Instead, local user accounts must be created by Administrators and Account Operators on the local computer. In turn, only Administrators and Account Operators at the domain level can create domain accounts. Domain accounts can then be used by Administrators or Account Operators on local computers in order to grant access to local resources.

User accounts, which include information such as the user's name, alias, password, and unique security identifier (SID), enable users to log on to the network or local computer and to access local and network resources.

To create, delete, and manage user accounts, administrators can use **User Accounts** in Control Panel, the **Local Users and Groups** snap-in to the Microsoft Management Console (if the user account is local to a particular computer), the **Active Directory Users and Computers** snap-in (if the account is to participate in a domain), or the command-line tools **Addusers.exe** or **Cusrmgr.exe**. For more information about these command-line tools, see "Managing Security Groups from the Command Line" later in this chapter. For more information about creating, deleting, and managing user accounts, see "Local Users and Groups" in Whistler Professional Help.

Types of Security Groups

Accounts are also members of security groups. Depending on the organizational environment, groups used to administer security can be defined by their scope, their purpose, their rights, or their role. The scope of a security group can be a single computer, a single domain, or multiple domains within a forest. In general, Windows 2000 and Whistler groups fall into one of several categories.

Computer local groups Computer local groups are security groups specific to a computer and are not recognized elsewhere in the domain. These groups are a primary means of managing rights and permissions to resources on a local computer.

Domain local groups Domain local groups are local to the domain in which they

are created, and thus can be given permissions and rights only to objects on computers within the domain.

Global groups Global groups, which are also created on domain controllers, are used for combining users who share a common access profile based on job function or business role. Global groups can contain user accounts from the same domain and other global groups from the same domain, and can be granted permissions to any Windows NT 4.0, Windows 2000, or Whistler computer in any domain in a forest.

Universal groups Universal groups are used in larger, multi-domain organizations where there is a need to grant access to similar groups of accounts defined in multiple domains in a forest. Universal groups are used only in multiple domain trees or forests that have a global catalog. They can contain groups from any Windows 2000 or Whistler domain, and can be used to grant access on any Windows 2000 or Whistler computers in any domain in the forest.

Special identities Special identities, which are also referred to as built-in security principals, apply to any account that is using the computer in a specified way, such as Anonymous and Remote logons. Unlike the other types of security groups, special identities do not have specific memberships that you can modify. You cannot view or modify the memberships of these identities, nor do you even see them when you administer groups. However, they are available for use when you assign rights and permissions to group members.

The scope of a group determines where in the network you are able to use the group and assign permissions, and the amount of network traffic the group creates. Using the most appropriate group for a task simplifies administration and, in a domain environment, reduces network traffic by reducing the amount of replication required.

Note

For more information about using domain local, universal, and global groups, see "Users, Computers, and Security Groups" in the *Distributed Services Guide* of the *Whistler Server Resource Kit*.

Computer Local Security Groups

Whistler Professional includes the following built-in computer local security groups:

- **Administrators**. Members of this group have total control of the local computer. They can create or delete user accounts and modify permissions for users and resources. Default members of the local Administrators group include the first account created on a clean installation, existing members of the local Administrators group in an upgrade, and members of the domain Administrators group.

> **Tip**
> Limit the membership of the Administrators group. The greater the number of members in the Administrators group, the greater the number of accounts that a hacker can potentially use to gain access to a computer.

- **Power Users**. By default, members of this group have Read/Write permission to other parts of the system in addition to their own profile. The Power Users group is an insecure group that is provided primarily for backward compatibility. Members of this group can run non-certified applications that will not run successfully under the secure Users group. When computers running Windows NT 4.0 or earlier are upgraded to Whistler Professional, all users are made members of the Power Users group.

- **Users**. By default, members of the Users group have Read/Write permissions only to their own profile. As a result, the Users group is secure to the extent that members cannot modify other users' data or system-wide settings, and they cannot run viruses or Trojan horse applications that affect the operating system or other users of the operating system. Typically, only applications that are certified for Windows 2000 or Whistler run successfully under the secure Users context.

> **Important**
> Users might need to have Power User privileges if you are running applications that have not been certified for use with Windows 2000. Test all of your applications at the privilege levels of the users who need to run them.

- **Guests**. By default, members of Guests group are denied access to the application and system event logs. Otherwise, members of the Guests group have the same access rights as members of the Users group. This allows occasional or one-time users to log on to a workstation's built-in Guest account and be granted limited abilities. Members of the Guest group can also shut down the system.

> **Note**
> The Guest account, which is a member of the Guests group by default, is not considered to be an authenticated user. Therefore, when logged on interactively, the Guest account is a member of both the Guests group and the Users group. However, when logged on over the network, the Guest account is not a member of the Users group.

- **Backup Operators**. Members of this group can back up and restore files on the computer, regardless of the permissions that protect those files. They can also log on to the computer and shut it down, but they cannot change security settings.
- **Replicators**. Members of this group are allowed to replicate files across a domain.
- **Network Configuration Operators**. Members of this group have limited administrative privileges that allow them to configure networking features, such as IP address assignment.
- **HelpServicesGroup**. Members of this group can utilize helper applications to diagnose system problems. This account can be used by members of Microsoft Help and Support Services to access the computer from the network and to log on locally.
- **Remote Desktop Users**. Members of this group have the right to log on locally.

Domain local versions of all of these groups — plus a number of additional server-specific built-in groups — are included on domain controllers.

Users' access to the local computer and network depends primarily on the computer local and domain local security groups to which their account belongs. In other words, users' accounts identify who they are, and in some cases permissions and restrictions are set on

an individual basis. However, the security groups to which the user belongs are primarily responsible for determining what permissions and restrictions govern their activities on the local computer and on the network.

For more information about the rights and permissions of computer local groups, see "User Rights" later in this chapter.

Special Identities

Special identities apply to any account that is using the computer in a specified way. Special identities allow you to configure security based on the manner in which a resource is being accessed.

The following special identities apply to any user account that is using the computer in a specified way:

Authenticated User. Any user, excluding the Guest account, who is authenticated locally, by a trusted domain controller. This identity provides users with the rights necessary to operate the system as an end user. (The Guest account is never treated as an Authenticated User.)

Interactive. Any user that logs on locally.

Anonymous Logon. Network logons for which credentials are not provided. Users cannot log on anonymously and interactively at the same time.

Creator Owner. The user account for the user who has created or taken ownership of a resource.

Dialup. Any user who accesses the computer over a dial-up connection.

Network. Any user that logs on over the network.

Everyone. All users who access the computer, including Guests and Users from other domains. By default, Everyone includes Authenticated Users and Guests, but not Anonymous logons.

The following special identities apply to any non-human user that is using the computer in a specified way:

Batch. Any batch process that is accessing a resource on the computer.

Service. Any service.

System. The operating system

Special identities are used to manage the rights and restrictions that apply to users based on the type of logon session they have initiated. For example, suppose there is a file or share that you want Alice to be able to access — but only when she is logged on interactively. You can accomplish this by allowing Alice access to the resource but denying access to requests accompanied by access tokens that include the Dialup special identities.

Well-Known SIDs

SIDs are associated with a user's account, security groups, and special identities. Most of these SIDs are unique. However, the values of some specific SIDs are constant across all systems. These are called well-known SIDs because they identify generic users or generic groups. For example, well-known SIDs identify the following users and groups:

- **Everyone (S-1-1-0).** The generic group Everyone automatically includes everyone who uses the computer, even users with anonymous guest accounts. The identifier authority value for this SID is 1 (World Authority). It has only one subauthority value, 0 (Null RID).

- **Creator Owner (S-1-3-0).** The generic user Creator Owner is a placeholder in an inheritable ACE. When the ACE is inherited, the system replaces the SID for Creator Owner with the SID for the object's current owner. The identifier authority value for this SID is 3 (Creator Authority). It has only one subauthority value, 0 (Null RID).

- **Self (S-1-5-10).** The generic user Self is a placeholder in an ACE on a User, Group, or Computer object in Active Directory. When you grant permission to Self, you grant it to the security principal represented by the object. During an access check, the operating system replaces the SID for Principal Self with the SID for the security principal represented by the object. The identifier authority for this SID is 5 (NT Authority). It has only one subauthority value, 10 (Self RID).

For information about other well-known SIDs, see the appendix "Well-Known Security Identifiers" in the *Distributed Services Guide*.

Managing Permissions by Nesting Groups

Nesting groups, or adding groups to other groups, can reduce the number of permissions that need to be assigned to users or groups individually. As you assign members of your organization to global groups in order to apply security settings based on a user's job or business unit, you can nest the groups into the Users and Power Users groups, and in this way apply the security settings that are inherent to Users and Power Users to the members of the global groups contained within them.

For example, Alice and other employees in the Accounting department can be added to a group that is specific to that department. An Administrator responsible for the Accounting department can control the membership of this group. The Administrator can assign organization-wide security permissions to these users by making the Accounting department security group a member of the Users domain local group. The Administrator thus only needs to configure the Accounting department security group to allow members access to the resources specific to the Accounting department.

This also facilitates the management of employees who are reassigned within an organization. It is much easier, for example, to move a user from the Accounting security group to the Marketing group than it is to reconfigure the many ACEs and ACLs required to permit the user to access the resources needed to perform the new job, and remove access to resources the user no longer needs.

Nesting Groups in Domain Environments

The process of creating groups across domains involves the following steps:

- Administrators in each domain create global groups and add user accounts that have the same resource requirements to the global groups.
- A domain administrator creates a domain local group for each resource that exists within a domain, such as file shares or printers, and then adds the appropriate global groups from each domain to this domain local group.
- A domain administrator assigns the appropriate permissions for the resource to the domain local group. Users in each global group receive the required permissions because their global group is a member of the domain local group.

Effectively nesting groups in a multi-domain environment reduces network traffic between domains and simplifies administration in a domain tree. The extent to which you can use nesting in your organization depends on whether you are operating in mixed mode or in native mode. In mixed mode, only one type of nesting is available: global groups can be members of domain local groups. Universal groups do not exist in mixed mode. In native mode, multiple levels of nesting are available. The nesting rules for group memberships for Windows 2000 and Whistler are listed in Table 18.1.

Table 18.1 Nesting Rules for Group Memberships

Group Scope	Can contain	Can be a member of
Domain Local Group	User accounts and universal and global groups from any trusted domain. Domain local groups from the same domain.	Domain local groups in the same domain.
Global Group	User accounts and global groups from the same domain.	Universal and domain local groups in any domain. Global groups in the same domain.
Universal Groups	User accounts, and universal and global groups from any domain.	Domain local or universal groups in any domain.

Working With Access Control Lists

Configuring ACLs for resource groups or security groups, and adding or removing users or resources from the appropriate groups when your organization changes, makes user permissions easier to control and audit, and reduces the need to change ACLs as frequently.

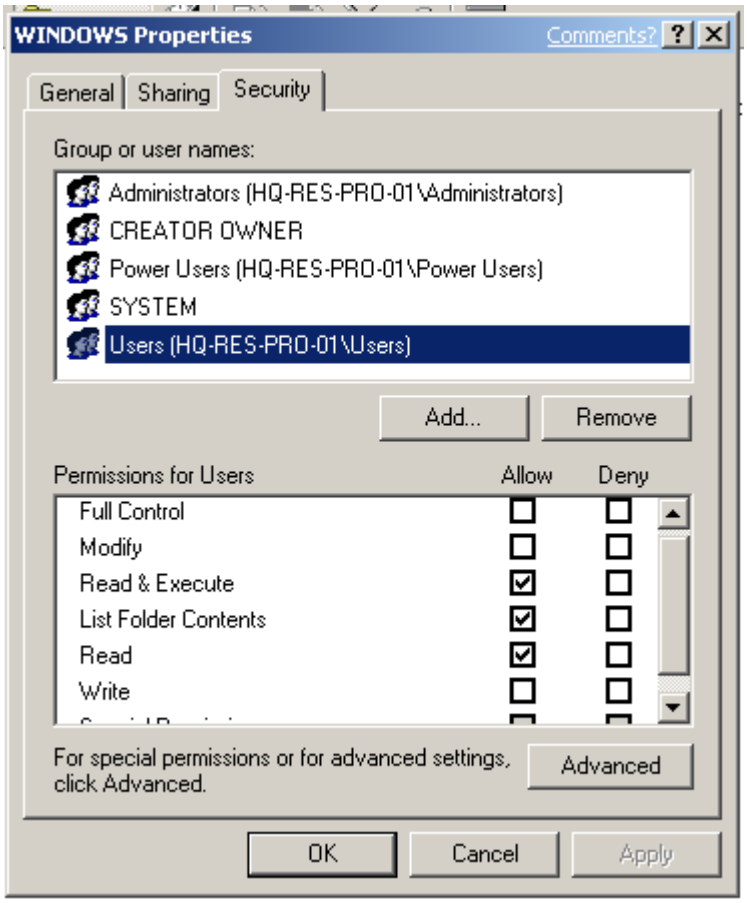
There are two types of ACLs — a Discretionary Access Control List (DACL), which identifies the users and groups that are allowed or denied access, and a System Access Control List (SACL), which controls how access is audited. For more information about the use of SACLs, see "Auditing and Evaluating Access Control" later in this chapter.

Viewing ACLs

The access control list for an object is generally found in the **Security** tab of the object's property sheet. This is the ACL Editor. This tab lists the groups and users that have access to this object, and provides a summary of the permissions allowed to each group.

Figure 18.2 illustrates an ACL Editor with a number of ACEs exposed.

Figure 18.2 Security Properties page for a Windows folder



The **Group or User names** box lists the security principals that have permissions assigned for this resource. The **Permissions for** box lists the permissions allowed or denied for the security principal highlighted in the **Group or User names** box. The **Add** and **Remove** buttons allow you to add new security principals for this resource or to delete existing principals from the list.

Note

Generally, the **Group or User names** box will include the resolved network names for the security principal. If the name does not resolve — if the computer is disconnected from the network, for example — the user or group's SID might appear instead.

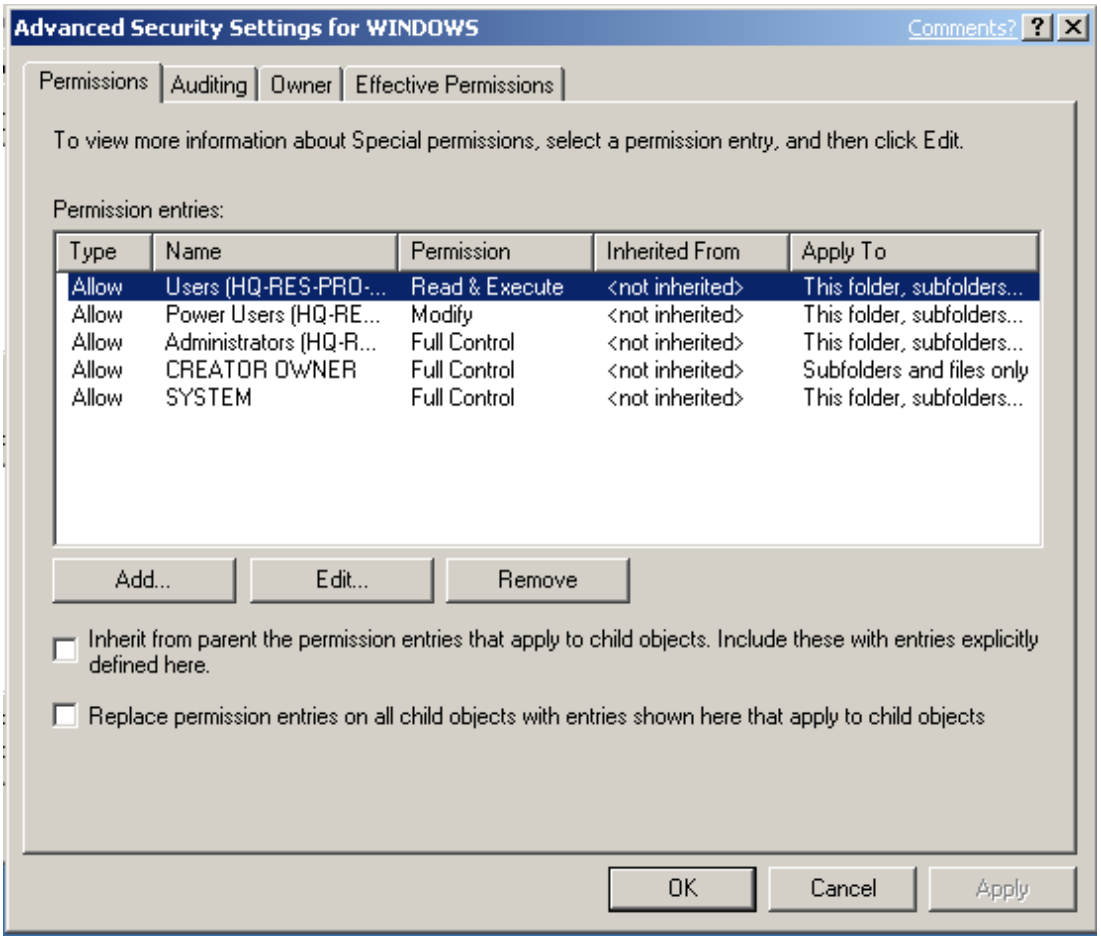
To view the ACL Editor on your system

1. Right-click an object such as a file, folder, or printer, and select **Properties**.
2. Click the **Security** tab.

Clicking the **Advanced** button opens the **Advanced Security Settings** page, which provides additional information about the Permissions that apply to a user or group.

Figure 18.3 shows an example of an Advanced Security Settings page.

Figure 18.3 Advanced Security Properties for a Windows folder



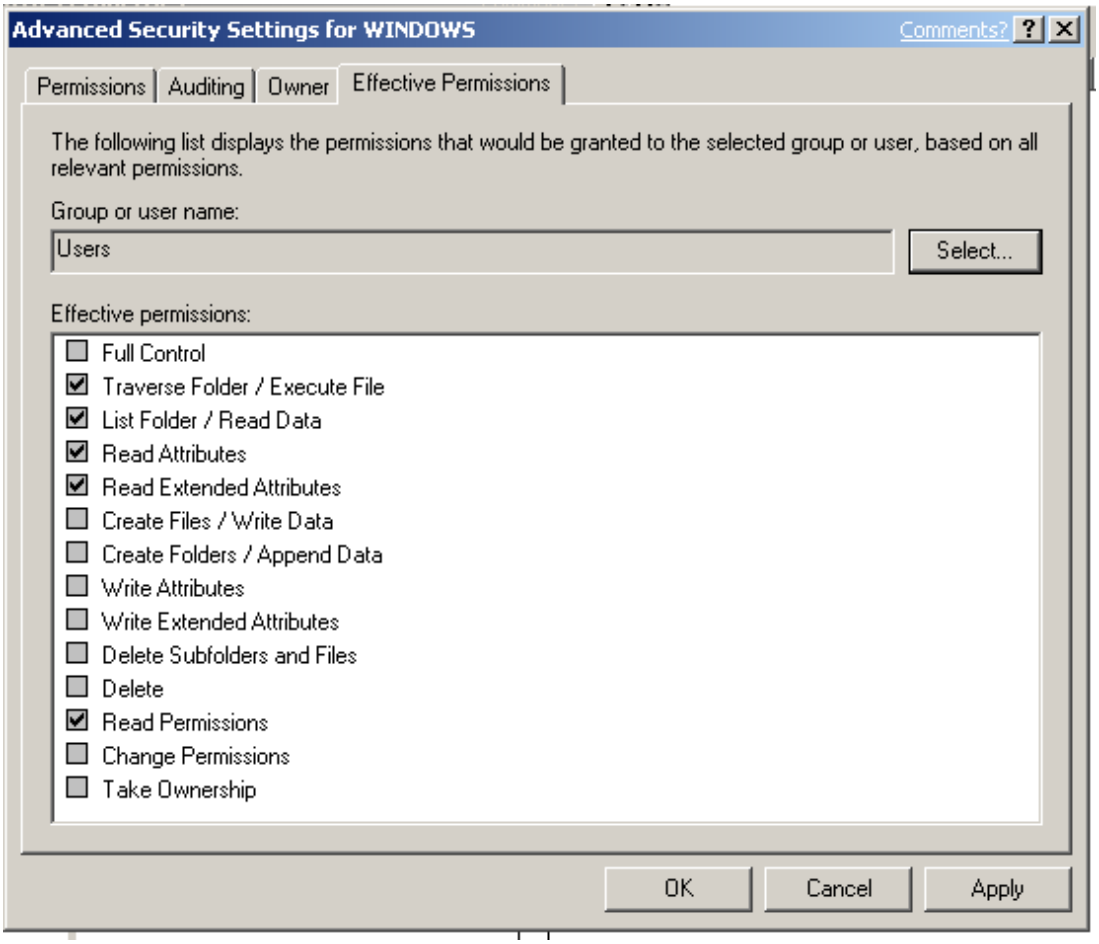
The **Advanced Security Settings** page allows you to utilize more advanced features for granting permissions, such as:

- Listing more detailed permissions that apply to each user or group.
- Defining or modifying access inheritance options for the object and any child objects.
- Auditing attempts to access the object.
- Viewing and modifying ownership information for the object and any child objects.
- Viewing effective permissions.

> **Note**
> If settings have been inherited from a parent object rather than explicitly defined on the object you are accessing, you might have to go back to the source ACL in order to change access control settings on the child container.

The **Permissions** tab shows only the permissions that the user has been granted directly. A new advanced option in the Whistler ACL editor, the **Effective Permissions** tab, allows you to see all of the permissions that apply to a security principal for a given object, including the permissions derived from memberships in security groups. These effective permissions regulate which users can gain access to the object and in what manner. The Effective Permissions tab is illustrated in Figure 18.4.

Figure 18.4 The Effective Permissions tab



To view the Effective Permissions for a user or group

1. Press the **Select . . .** button.
2. Type the name of the user, group, or security principal for which you would like to view the Effective Permissions. If necessary, press the **Look For . . .** button, select whether you are searching for a built-in security principal, computer, group, or user, and click **OK**.
3. Click **OK**.

Tip

If the security principal is network based, you might need to type in the domain name together with the group name, such as reskit\users.

Access Control Entries

Access Control Lists contain a wide variety of ACEs that can be viewed on the Effective Permissions page. All ACEs include the following access control information:

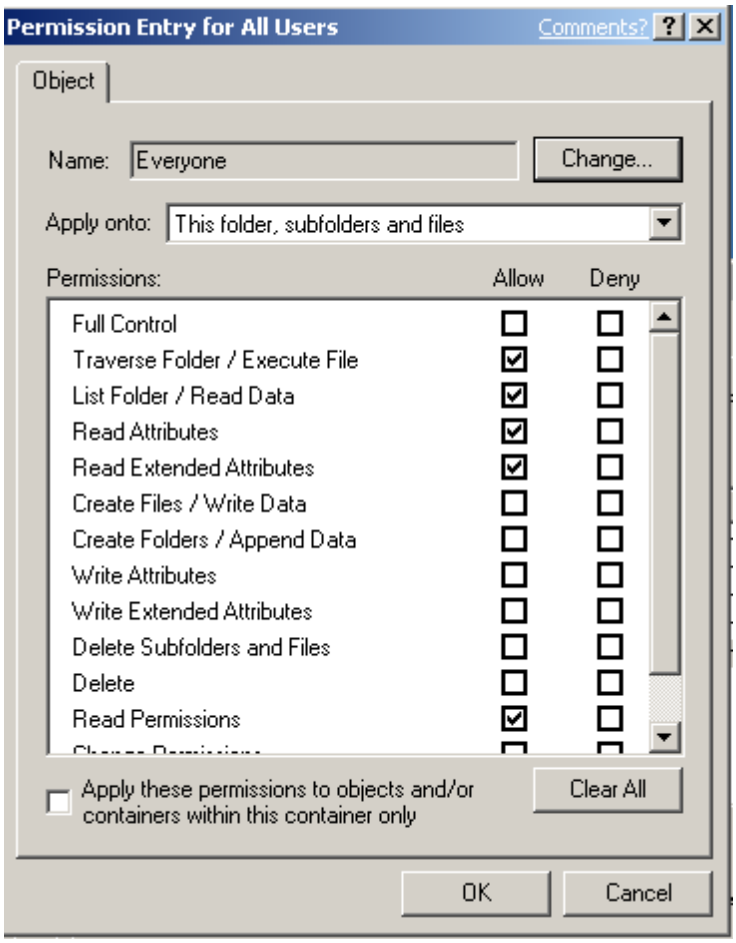
- One SID that identifies a user or group, such as Alice, the Accounting department, or users in the Denver office.
- A list of permissions that specify access rights, such as List Folder/Read Data.
- Inheritance information that determines whether new files created in a particular folder will utilize the same access permissions as the parent folder.
- A flag that indicates the type of ACE.

To view a specific ACE

1. Navigate to the **Advanced Security Settings** page for the file, folder, or object.
2. Select and double-click one of the entries in the **Permission Entries** box.

Figure 18.5 illustrates the ACE for the System special identity on the Documents and Settings\All Users folder.

Figure 18.5 ACE for the All Users folder



How Access Control is Applied to New Objects

The operating system uses the following guidelines to set the DACL in the security descriptors for most types of new securable objects:

1. The new object's DACL is the DACL from the security descriptor specified by the creating process. The operating system merges any inheritable ACEs from the parent object into the DACL.

2. If the creating process does not specify a security descriptor, the operating system builds the object's DACL from inheritable ACEs in the parent object's DACL. For example, in the case of a new file, this might be the inheritable ACEs from the folder in which the file is being created.

3. If the parent object has no inheritable ACEs, for example if the file is being created in the root directory, the operating system asks the object manager to provide a default DACL.

4. If the object manager does not provide a default DACL, the operating system checks the subject's (the user, for example) access token for a default DACL.

5. If the subject's access token does not have a default DACL, the new object is assigned no DACL, which allows Everyone unconditional access.

> **Important**
> Failure to set DACLs or setting DACLs improperly can have undesirable consequences. For example, an empty DACL, where neither Allow nor Deny has been configured, denies access to all accounts. On the other hand, a null DACL gives full access to all accounts.

The operating system uses similar guidelines to set the SACL in the security descriptors for new securable objects.

Modifying Inheritance of Permissions

Inheritance is one of the primary tools for managing access control. By default, permissions assigned to a parent folder are inherited by the subfolders and files that are contained in the parent folder. You can block inheritance, however, so that permission changes made to parent folders will not affect child folders and files.

To block permission changes made to parent folders from affecting child folders and files

1. Open the **Advanced Security settings** page for the file or folder.

2. Click the **Permissions** tab.

3. Unselect **Inherit from parent the permission entries that apply to child objects**. **Include these with entries explicitly defined here.**

4. Click **OK**.

Permissions can also be denied. By denying a user or group permission to a folder or file, you are denying a specific level of access regardless of the other permissions assigned to the user or group. Even if a user has access permissions to the file or folder as a member of

one group, denying permission to the user as a member of a second group blocks any other permissions the user has.

Managing Ownership Permissions

You can take ownership of a resource if you are a member of the Administrators group. It is important for administrators to take full ownership or reassign ownership for key resources, so that if an employee creates a resource, such as a file share, for a project and then leaves the organization, that resource does not become inaccessible.

To view the ownership information associated with a resource

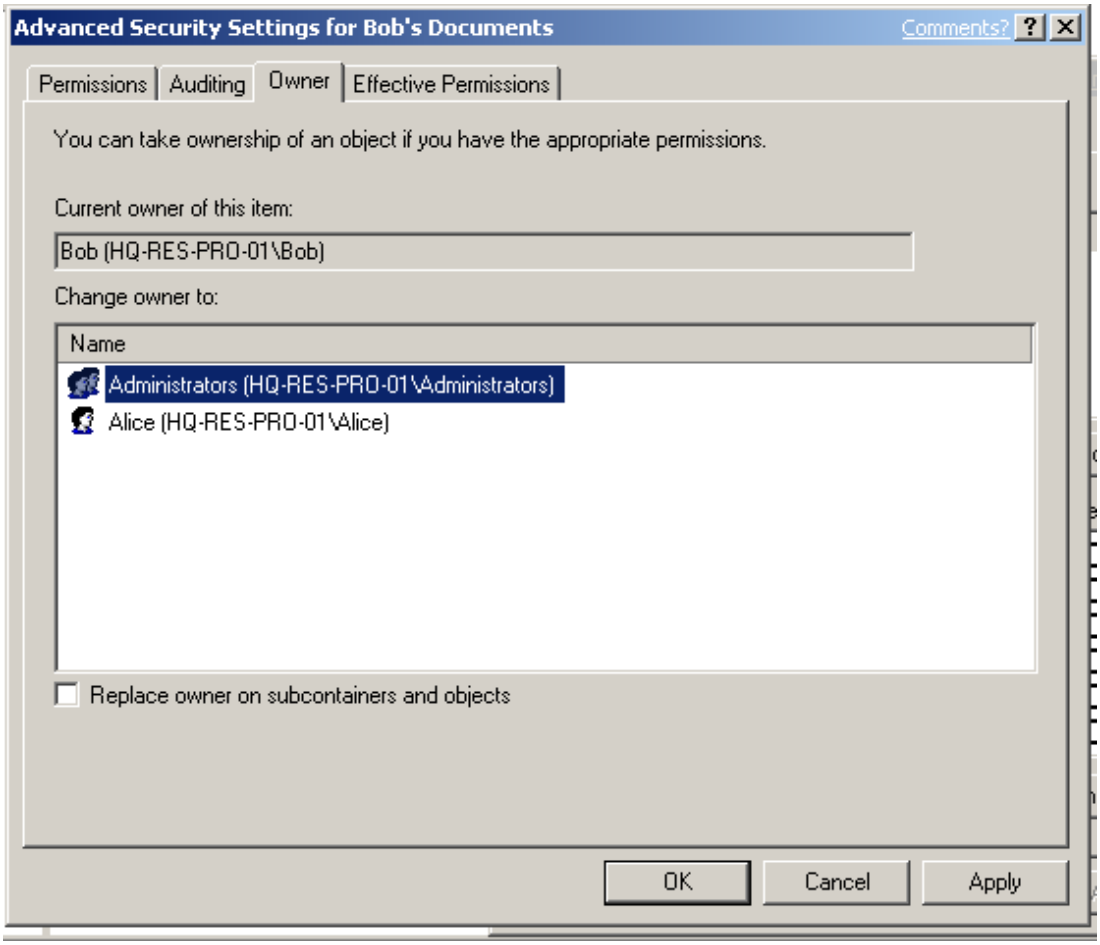
1. Right-click the file or folder and select **Properties** from the secondary menu.
2. From the **Security** tab, click the **Advanced** button to view the **Advanced Security settings** of the resource.
3. Click the **Ownership** tab.

Note

You must have Read permissions on the object in order to view ownership data.

Figure 18.6 illustrates the **Ownership** tab.

Figure 18.6 The Ownership tab



Every object has an owner, usually the user who created the object. The owner has an implied right to Allow or Deny other users permission to use the object. This right cannot be withdrawn. Owners can give other users permission to Change Permissions (WRITE_DAC). This permission, unlike the owner's inherent right, can be withdrawn.

By default, a new object's owner is the security principal identified as the default owner in the access token attached to the creating process. When an object is created, the SID stored in the access token's Owner field is copied to the security descriptor's Owner field. The default owner is normally an individual — the user who is currently logged on. The only exceptions occur when the user is a member of the Administrators group. In this case, the Owner field in the user's access token contains the SID for the group, not the SID for the individual user account. The assumption is that administrative accounts are used only to

administer the system and not for any individual purpose. As a result, objects created by one administrator can be managed by other administrators in the same group.

If an administrative group such as Administrators owns an object, all members of the group share the owner's inherent right to change permissions for the object. For example, suppose Alice logs on to an account in the Administrators group, creates a file, and then denies Bob permission to modify it. Because Alice is a member of the Administrators group, the group owns the file. If Bob is also a member of the Administrators group, he automatically has Change Permissions authority and can give himself permission to modify the file — despite Alice's effort to prevent him from modifying it.

Owners of NTFS objects can allow another user to take ownership by giving that user Take Ownership permission. In addition, certain users can take ownership without having permission if they have been assigned the **Take ownership of files or other objects** (SeTakeOwnershipPrivilege) privilege. By default, this privilege is assigned only to the Administrators group.

When a user takes ownership of an object, the default owner SID in the user's access token is copied to the owner field of the object's security descriptor. If a member of the Administrators group takes ownership, the default owner is the group, not the individual user. Therefore any member of the Administrators group can exercise ownership of the resource.

To determine the owners of objects in a share or folder

- At the command line, type:

  ```
  dir /q <share or folder name>
  ```

Default Permissions

Whistler Professional offers a very fine degree of security control over access to a wide variety of objects. A local file folder, for example, has 13 available permissions, beginning with Read, Write, Modify, and Delete. Both basic and advanced permissions are available for files and folders.

Basic File and Folder Permissions

The number and type of permissions that are available for any object depend on the context of the object. For example, the following permissions are available for folders on NTFS partitions:

Read. Allows a user to see the files and subfolders in a folder and view folder attributes, ownership, and permissions.

Write. Allows a user to create new files and subfolders with the folder, change folder attributes, and view folder ownership and permissions.

List Folder Contents. Allows a user to see the names of files and subfolders in the folder.

Read and Execute. Gives a user the rights assigned through the Read permission and the List Folder Contents permission. It also gives the user the ability to traverse folders. Traverse folders rights allow a user to reach files and folders located in subdirectories even if the user does not have permission to access portions of the directory path.

Modify. Gives a user the ability to delete the folder and perform the actions permitted by the write and read/execute permissions.

Full Control. Allows a user to change permissions, take ownership, delete subfolders and files, and perform the actions granted by all other permissions.

The following basic permissions apply to files on NTFS partitions:

Read. Allows a user to read a file and view file attributes, ownership, and permissions.

Write. Allows a user to overwrite a file, change file attributes, and view file ownership and permissions.

Read and Execute. Gives a user the rights required to run applications and perform the actions permitted by the read permission.

Modify. Gives a user the ability to modify and delete a file and perform the actions permitted by the Write and Read/Execute permissions.

Full Control. Allows a user to change permissions, take ownership, delete subfolders and files, and perform the actions granted by all other permissions.

Note

Share permissions for NTFS volumes work in combination with file and directory permissions. By default, the permissions for a new share on an NTFS partition allow Everyone/Full Control. Using Full Control permission for Everyone on all NTFS shared directories is the easiest way to manage NTFS file security. This way, you have to manage only the file and directory permissions.

Advanced File and Folder Permissions

A number of additional permissions are available by clicking **Advanced** on the ACL Editor page. These include:

Traverse Folder/Execute File. Allows or denies moving through folders to reach other files or folders, even if the user has no permissions to the folders being traversed (the permission applies only to folders). Traverse Folder takes effect when a group or user is not granted the Bypass Traverse Checking user right in the Group Policy Snap-In. (By default, the Everyone group is given the Bypass Traverse Checking user right.) The Execute File permission allows or denies running program files (the permission applies only to files).

Note

Setting the Traverse Folder permission on a folder does not automatically set the Execute File permission on all files within that folder.

List Folder/Read Data. Allows or denies viewing filenames and subfolder names within the folder (the permission applies only to folders). The Read Data permission allows or denies viewing data in files (the permission applies only to files).

Read Attributes. Allows or denies viewing the attributes of a file or folder (for example, the read-only and hidden attributes). Attributes are defined by NTFS.

Read Extended Attributes. Allows or denies viewing the extended attributes of a file or folder. Extended attributes are defined by programs and can vary by program.

Create Files/Write Data. Allows or denies creating files within the folder (the permission applies only to folders). Also, the Write Data permission allows or denies making changes to the file and overwriting existing content (the permission applies only to files).

Create Folders/Append Data. Allows or denies creating folders within the folder (the permission applies only to folders). The Append Data permission allows or denies making

changes to the end of the file but not changing, deleting, or overwriting existing data (the permission applies only to files).

Write Attributes. Allows or denies changing the attributes of a file or folder.

Write Extended Attributes. Allows or denies changing the extended attributes of a file or folder. Extended attributes are defined by programs and can vary by program.

Delete Subfolders and Files. Allows or denies deleting subfolders and files, even if the Delete permission has not been granted on the subfolder or file.

Delete. Allows or denies deleting the file or folder. If you don't have Delete permission on a file or folder, you can still delete it if you have been granted Delete Subfolders and Files permission on the parent folder.

Read Permissions. Allows or denies reading permissions of a file or folder, such as Full Control, Read, and Write.

Change Permissions. Allows or denies changing permissions of the file or folder, such as Full Control, Read, and Write.

Take Ownership. Allows or denies taking ownership of a file or folder. The owner of a file or folder can always change permissions on it, regardless of any existing permissions that protect the file or folder.

It is not necessary to configure advanced permissions independently of basic permissions, because many of the advanced permissions are already configured when you select certain basic permissions. For example, Table 18.2 illustrates the links between basic and advanced permissions for folders.

Table 18.2 Advanced Folder Permissions

Special Permissions	Full Control	Modify	Read & Execute	List Folder Contents	Read	Write
Traverse Folder/Execute File	Yes	Yes	Yes	Yes		
List Folder/Read Data	Yes	Yes	Yes	Yes	Yes	
Read Attributes	Yes	Yes	Yes	Yes	Yes	

Read Extended Attributes	Yes	Yes	Yes	Yes	Yes	
Create Files/Write Data	Yes	Yes				Yes
Create Folders/Append Data	Yes	Yes				Yes
Write Attributes	Yes	Yes				Yes
Write Extended Attributes	Yes	Yes				Yes
Delete Subfolders and Files	Yes					

Table 18.3 illustrates the links between basic and advanced permissions for files.

Table 18.3 Advanced File Permissions

Special Permissions	Full Control	Modify	Read & Execute	Read	Write
Traverse Folder/Execute File	Yes	Yes	Yes		
List Folder/Read Data	Yes	Yes	Yes	Yes	
Read Attributes	Yes	Yes	Yes	Yes	
Read Extended Attributes	Yes	Yes	Yes	Yes	
Create Files/Write Data	Yes	Yes			Yes
Create Folders/Append Data	Yes	Yes		Yes	
Write Attributes	Yes	Yes			Yes
Write Extended Attributes	Yes	Yes			Yes
Delete	Yes	Yes			
Read Permissions	Yes	Yes	Yes	Yes	Yes

Change Permissions	Yes				
Take Ownership	Yes				
Synchronize	Yes	Yes	Yes	Yes	Yes

Note

File and folder permissions are only available with the NTFS file system. File and folder security permissions are not available with the FAT or FAT32 file systems.

Applying Folder and Share Permissions at Setup

Default NTFS file permissions for the installation partition are applied during setup by the Security Configuration Manager using the following security templates, which are located in the %windir\inf directory:

- Defltwk.inf — for a clean installation.
- DWUp.inf — for an upgrade.

Upgrades from Win9x platforms are treated as clean installs.

The Security Configuration Manager also secures the root directory during setup if the current root security descriptor grants Everyone Full Control. This is a change from previous releases of Windows NT and provides increased security for non-Windows directories that are created off of the root. Because of the ACL inheritance model, any non-Windows subdirectories that inherit permissions from the root directory will also be modified during setup. The new Whistler root ACL (also implemented by Format and Convert) is as follows:

- Administrators, System: Full Control (Container Inherit, Object Inherit)
- Creator Owner: Full Control (Container Inherit, Object Inherit, Inherit Only)
- Everyone: Read\Execute (No Inheritance)
- Users: Read\Execute (Container Inherit, Object Inherit)
- Users: Create Directory (Container Inherit)
- Users: Add File (Container Inherit, Inherit Only)

Security Configuration Manager also creates a security template called setup security.inf in the %windir%\security\templates directory, which can be used to reapply default security settings in the future.

Using Command Line Tools to Work with ACLs

- The ACL Editor, the basic user interface for viewing and modifying ACLs and ACEs, is not available for configuring security for all types of objects on a network or Whistler Professional computer. For example, it cannot be used to configure local groups and services, nor is it the best available tool for performing all ACL-related administrative tasks. Instead, you can use Cacls.exe, a command-line tools that allows you to work with ACLs.

> **Caution**
> This tool is designed for use by administrators. If a user does not have sufficient privileges to use this tool, some actions might fail or generate error messages.

Cacls.exe

Cacls.exe can be used to display or modify access control lists (ACLs) for one or more files at time. It includes options that can be used to grant (/g), revoke (/r), replace (/p), or deny (/d) specific user access rights. For example:

To grant a user an access right

- At the command line, type:

```
cacls filename [/g user:perm]
```

Managing User Rights Through Security Groups

The rights granted to a user are based on the user's security group memberships. For this reason, a significant portion of Whistler operating system security is defined by the default access permissions granted to the Administrators, Power Users, and Users groups. If you already have a managed user environment, or if you want to move to a managed user environment, consider the capabilities and restrictions that apply to each of these security

groups. Also, determine which of your users require higher levels of permissions, and which users require fewer permissions.

Administrators

Administrators have unlimited access. The default Whistler security settings do not restrict Administrative access to any registry or file system object. Administrators can perform any and all functions supported by the operating system. Any right that the Administrator does not have by default, they can grant to themselves.

Important

When a user is logged on as an Administrator, his or her security context can be used by potential hackers to perform detrimental actions on the local computer, or, in the case of a network administrator, on the network. Impress upon all members of the Administrators group the importance of minimizing the amount of time that they are logged on with these privileges.

Administrative access to the system is ideally used only to:

- Install the operating system and components (including drivers for hardware, system services, and so forth).
- Install service packs and hot fixes.
- Install Windows updates.
- Upgrade the operating system.
- Repair the operating system.
- Configure critical machine-wide operating system parameters.

In some cases, administrative accounts must also be used to install and run legacy Windows-based applications.

Users

Unlike Administrators, Users have limited access on the system. User security settings are designed to prohibit members of the Users group from compromising the integrity of the operating system and installed applications. Users cannot modify machine-wide registry settings, operating system files, or program files, and they cannot install applications that can be run by other Users. Users cannot access other users' private data. The best way to increase the security and manageability of the operating system is to make all end-users

members of the Users group only, and deploy only applications that are certified for Windows 2000 and Whistler.

Unfortunately, many legacy applications were not designed with operating system security in mind, and as a result, members of the Users groups cannot run them. Members of the Power Users group, however, can run such applications.

Applications that comply with the Windows 2000 Application Specification can successfully run under a Users context.

Power Users

Power Users have less system access than Administrators but more than Users. The default Windows 2000 and later security settings for Power Users are backward-compatible with the default security settings for Users in the Windows NT 4.0 operating system. This allows Power Users to run legacy applications that are not certified for Windows 2000 and Whistler and therefore cannot be run under the more secure Users context.

Power Users can perform many system-wide operations, such as changing system time and display settings, creating user accounts and shares, and installing printers. Power Users also have Modify access to:

- HKEY_LOCAL_MACHINE \Software
- Program files
- %***windir***%
- %***windir***%\system32

Although Power Users have Modify access to the *windir* and %*windir*%\system32 directories, they have Read-only access to the files that are installed in these directories during Whistler Professional text-mode setup. This allows non-certified applications to write new files into the system directories but prevents Power Users from modifying the Whistler Professional system files.

While Power Users have the permissions necessary to install most applications, not all application installations will succeed. For example, many applications check for explicit membership in the Administrators group before installing. Other applications attempt to replace operating system files, which Power Users cannot do. Finally, because Power Users cannot install services, they cannot install applications that have a service component.

Like Users, Power Users are not allowed to access data stored in other users' profiles.

Security Group Upgrade from Windows NT 4.0

In the case of an upgrade from Windows NT 4.0 to Windows 2000 or Whistler Professional, existing Users automatically become members of the Power Users group. This is because the permissions that apply to Users in Windows 2000 and Whistler Professional are more restrictive than the permissions that apply to Users in Windows NT 4.0. Because the privileges granted to the Users group in Windows 2000 and later are more restrictive than those granted to Users in Windows NT 4.0, certain applications might not run for users who are members of the Users group. By placing Windows NT 4.0 Users in the Whistler Power Users group, these users can continue to run non-certified applications.

From a security standpoint, deploying certified applications and placing users in the Users group only is preferred. The default access control settings for the Users group on NTFS systems prevent users (or malicious applications run by users) from compromising the operating system or other users' data. Similarly, it is best if administrative personnel log on with administrative privileges only when it is necessary to perform administrative tasks. Through the use of the RunAs command, administrative personnel can run under a normal User context, then launch administrative programs in an Administrator context without having to log off.

Note

If you need to run non-certified applications but do not want to use the Power Users group, the Compatible security template can be used to open up permissions for Users in a manner that is consistent with the access control requirements of most legacy applications. For more information about the Compatible template, see "Security Templates" later in this chapter.

Managing Anonymous Logons

It is recommended that you do not grant access to Anonymous Logons unless you have specific reasons for doing so. To help you implement this restriction, when a Windows 2000 system is upgraded to Whistler, resources with access control lists that grant access to the Everyone group (and not explicitly to the Anonymous Logon group) will no longer be available to Anonymous users.

In most cases, this is an appropriate restriction on Anonymous access. However, you might need to permit Anonymous access in order to support pre-existing applications that require it. In this case, you can explicitly add the Anonymous Logon security group to the access control lists for the specific resources and grant Anonymous users the right to access this computer over the network.

In some situations, however, it might be difficult or impossible to determine which resource on the computer running Whistler Professional must grant Anonymous access, or to modify the permissions on all of the necessary resources. If this is the case, you might need to force the computer running Whistler to include the Everyone group in the Anonymous Logon security token. To support this, Whistler introduces a new registry value, EveryoneIncludesAnonymous, which you can use to switch between the default Whistler behavior (the Everyone security group does not include the Anonymous Logon security group) and the Windows 2000 behavior (the Everyone security group includes the Anonymous Logon security group).

The EveryoneIncludesAnonymous registry value can be set either through the **Let Everyone permissions apply to anonymous users** local security setting or by editing the registry directly. The security setting can be set to either **Enabled** (DWORD 0x1) or Disabled (DWORD 0x0). The default setting is **Disabled**.

> **Caution**
> Do not edit the registry directly unless you have no alternative. The registry editors bypass standard safeguards, allowing settings that can degrade performance, damage your system, or even require you to reinstall Windows. You can safely alter most registry settings by using the programs in Control Panel or Microsoft Management Console (MMC). If you must edit the registry directly, back it up first. Read the Registry Editor Help for more information.

To set the EveryoneIncludesAnonymous registry value using local security settings

1. Click **Start**, point to **Programs**, point to **Administrative Tools**, and then click either **Local Security Policy** or **Domain Security Policy** (on domain controllers only)**.**

2. Under **Security Settings**, double-click **Local Policies**, and then click **Security Options**.

3. Right click **Let Everyone permissions apply to anonymous users**, and then click **Properties**.

4. To allow Anonymous users to be members of the Everyone security group, click **Enabled.** To revoke the inclusion of the Everyone security group security identifier in the Anonymous user's access token (the Whistler default), click **Disabled.**

To set the EveryoneIncludesAnonymous registry value directly

1. Click **Start,** click **Run,** type **Regedit**, and then click **OK**.

2. In the Registry Editor, navigate to and then click the **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa** registry key**.**

3. Right click the value **EveryoneIncludesAnonymous**, and then click **Modify**.

4. To allow anonymous users to be members of the Everyone security group, in **Value data**, type **1**. To revoke the inclusion of the Everyone security group security identifier in the anonymous user's access token (the Whistler default), in **Value data**, type **0**.

5. Close the Registry Editor.

Managing Network Logons

An increasing number of Whistler Professional systems are connected directly to the Internet rather than to domains. This makes proper management of access control (including strong passwords and permissions associated with different accounts) more critical than ever. To ensure security, the relatively anonymous access control settings commonly associated with open Internet environments need to be curtailed.

As a result, the default in Whistler Professional requires all network connections that utilize the Network special identity to use the Guest account, which by default is disabled. This change is designed to prevent hackers attempting to access a system across the Internet from logging on using a local Administrator account.

If you choose to override this default behavior, you need to make two key decisions:

1. Whether to allow network logons to access your system. You can accomplish this goal by enabling the Guest account.

2. Whether to allow network logons greater permissions than are associated with the Guest account. You can accomplish this by disabling the registry key that forces network logons to use the Guest account.

If you choose to allow one or both of these options, you must implement these changes along with your overall authentication and access control strategies, to ensure that outside users have only as much or as little access to the local system as you find to be appropriate.

Regardless of which option you choose, the link between the Network logon special identity and the Guest account does not affect the following:

- Interactive logons. This includes remote access using Terminal Server or Telnet, for example, which are essentially "remote" instances of interactive logon sessions.
- Network logons using domain accounts.
- Outbound connections. The access control settings of the computer you are attempting to access govern outbound connections.
- Upgraded systems. Configuration settings that were in effect before the upgrade will remain in effect.
- Most server-based applications. Server applications that use non-domain accounts to access workstations (which are extremely rare and not recommended) are the only applications that could be affected.

To ensure that remote administration of domain-based computers running Whistler Professional is possible, you must include a domain-based account in the local administrators group.

You can use the Group Policy snap-in to disable the registry setting that force Network logons to use the Guest account.

To allow network logons using non-Guest accounts

1. Open the Group Policy snap-in in the Microsoft Management Console and navigate to the **Security Settings** container:

Local Computer Policy\ Computer Configuration\ Windows Settings\ Security Settings\Local Policies\Security Options.

2. In the right-hand pane, select the policy **Network access: Force network logons using local accounts to authenticate as Guest,** and right-click. Select **Properties**.

3. Select **Disabled** and Click **OK**.

4. Exit the Group Policy snap-in.

You can accomplish the same goal directly by editing the registry using regedit.

> **Caution**
> Do not edit the registry directly unless you have no alternative. The registry editors bypass standard safeguards, allowing settings that can degrade performance, damage your system, or even require you to reinstall Windows. You can safely alter most registry settings by using the programs in Control Panel or Microsoft Management Console. If you must edit the registry directly, back it up first. Read the Registry Editor Help for more information.

To change the registry key forcing Network logons to use the Guest account:

1. At the **Run** command, type **Regedit** and click **Enter**.

2. Navigate to HKLM\System\CurrentControlSet\Control\LSA.

3. Select the **ForceGuest** registry value. Set ForceGuest=0 (Disabled). Exit Regedit.

You do not need to reboot in order for this registry change to take effect. For more information about using the Home Network Wizard to enable the Guest account for sharing files and folders, and to ensure that the personal firewall is properly configured, see Whistler Professional Help and "Connecting Remote Offices" in this book.

Security Group Creation in a Clean Install

In the case of a clean installation of Windows 2000 or Whistler Professional, new users are by default members of the Users group. By default, Administrators, System, and Creator Owner are given Full Control to all file system and registry objects that exist at the

beginning of GUI-mode setup. Users are explicitly granted Write access to specific locations, and Read-Only (or less) access to the rest of the system. Table 18.4 lists the default Write access permissions for Users in Whistler.

Table 18.4 Default Write Access Permissions for Users in Whistler

Object	Permission	Comment
HKEY_Current_User	Full Control	User's portion of the registry
%UserProfile%	Full Control	User's Profile directory
All Users\Documents	Modify	Shared Documents Location
All Users\Application Data	Modify	Shared Application Data Location
%Windir%\Temp	Synchronize, Traverse, Add File, Add Subdir	Per-Machine temp directory. This is a concession made for service-based applications so that Profiles do not need to be loaded in order to get the per-User temp directory of an impersonated user.
\ (Root Directory)	Not Configured during setup	Not configured during setup because the Windows 2000 ACL Inheritance model would impact all child objects including those outside the scope of setup.

Table 18.5 lists the default User rights for clean-installed workstations.

Table 18.5 User Rights for Clean-Installed Workstations

User Right	Default Workstation
Replace a Process-Level Token	
Generate Security Audits	
Logon as a Batch Job	
Backup Files and Directories	Administrators, Backup Ops
Bypass Traverse Checking	Administrators, Backup Ops, Power Users, Users, Everyone
Create a Pagefile	Administrators
Create Permanent Shared Objects	
Create a Token Object	
Debug Programs	Administrators

Increase Scheduling Priority	Administrators
Increase Quotas	Administrators
Logon Interactively	Administrators, Backup Ops, Power Users, Users, Guest[3]
Load and Unload Device Drivers	Administrators
Lock Pages in Memory	
Add workstations to the domain	
Access this computer from the network	Administrators, Backup Ops, Power Users, Users, Everyone
Profile a single process	Administrators, Power Users
Force shutdown from a remote system	Administrators
Restore files and directories	Administrators, Backup Ops
Manage audit and security logs	Administrators
Logon as a service	
Shutdown the system	Administrators, Backup Ops, Power Users, Users
Modify firmware environment variables	Administrators
Profile system performance	Administrators
Change system time	Administrators, Power Users
Take ownership of files or other objects	Administrators
Act as part of the OS	
Deny Interactive Logon	
Deny Batch Logon	
Deny Service Logon	
Deny Network Logon	
Remove Computer from a Docking Station	Administrators, Power Users, Users
Synchronize Directory Service Data	
Enable computer and user accounts to be trusted for delegation	

Permissions Associated with Special Identities

In Windows NT 4.0, the Everyone group is used as a catch-all for file system ACLs, registry ACLs, and User rights. An administrator cannot define who does and does not belong to the Everyone group. Instead, Windows NT 4.0 automatically controls the group membership so

that everyone is a member of the Everyone group. If an administrator wants more granular access control, the default ACLs have to be modified in order to remove the Everyone group and add the groups that the administrator can control.

In Windows 2000 and later, groups whose membership is automatically configured by the operating system, such as Everyone and Authenticated Users, are not used to assign permissions to file and registry objects. Only those groups whose membership can be controlled by an administrator — primarily Users, Power Users, and Administrators — are used to assign permissions. When users are members of a group, they automatically have the permissions that have been assigned to that group.

The users that constitute the default memberships in these groups are listed in Table 18.6.

Table 18.6 Default Group Memberships

Local Group	Default Workstation Members
Administrators	Administrator
Power Users	Interactive Users
Users	Authenticated Users

With a clean install of Windows 2000 and Whistler Professional, the Authenticated Users group and the Interactive group are added to the Users group. Thus, by default, any non-administrative user accessing a Windows 2000 or Whistler Professional–based system interactively is a member of the Users group. Because the Guest account and anonymous logons are not considered to be authenticated, these users do not receive User level access over the network.

On upgrades from Windows NT 4.0, the Interactive users group is added to the Power Users group. Because Windows 2000 and Whistler Professional Power Users have the same file system and registry permissions that Windows NT 4.0 Users have, Interactive users on Windows 2000 and Whistler Professional machines that were upgraded from Windows NT 4.0 can run any application that Windows NT 4.0 Users could run.

Deploying certified applications and then removing Interactive Users from the Power Users group secures a Windows 2000 or Whistler Professional–based workstation that was upgraded from Windows NT 4.0. In this way, non-administrators who log on will be subject to the secure permissions granted to the Users group without having to change any file or registry ACLs.

Managing Security Groups from the Command Line

The **Whoami** utility allows you to view the rights and permissions that apply to an individual user. This command-line tool returns the domain or computer name and the user name of the user who is currently logged on to the computer on which the tool is run, as well as the complete contents of the current user's access token. It displays the user name and security identifiers (SIDs), the groups and their SIDs, the privileges and their status (for example, enabled or disabled), and the logon ID.

To view an individual's rights and permissions

- At the command line, type:

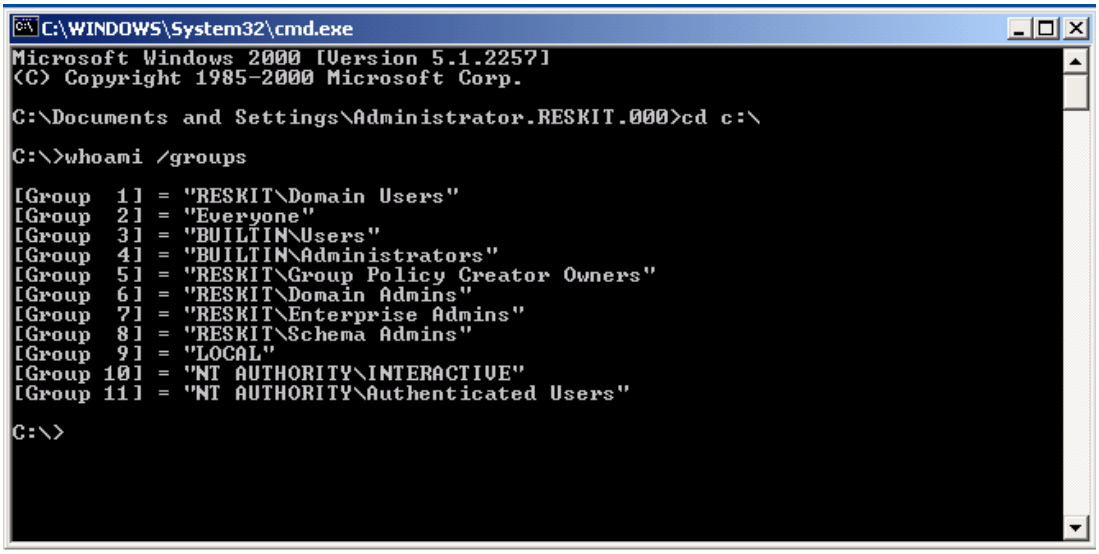
```
Whoami
```

You can use the following command-line options to customize the results you receive from whoami:

- **/ALL**. Displays all information in the current access token.
- **/USER**. Displays the user identified in the current access token.
- **/GROUPS**. Displays groups listed in the current access token.
- **/PRIV**. Displays privileges associated with the current access token.
- **/LOGONID**. Displays the logon ID used for the current session.
- **/SID**. Displays the SIDS associated with the current session (must be used in combination with the /USER, /GROUPS, /PRIV, or /LOGONID switches).
- **/NOVERBOSE**. Displays minimal information (must be used in combination with the /USER, /GROUPS, /PRIV, or /LOGONID switches).

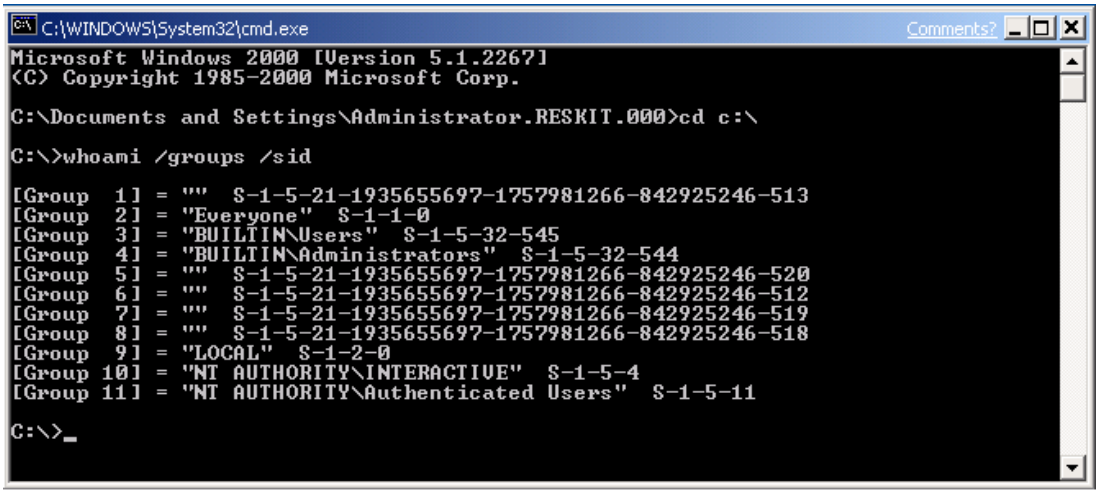
Figure 18.7 shows an example of the output you might receive when you run **whoami** to determine the groups that a user belongs to.

Figure 18.7 Sample output from the Whoami utility



Whoami also allows you to identify the unique security identifiers that are associated with a given logon session. For example, Figure 18.8 provides more detailed information about the logon session illustrated in Figure 18.7.

Figure 18.8 Results from a SID query using Whoami



Using Security Policy

Security Configuration Manager provides the Security Settings extension to Group Policy. With this extension, numerous security-relevant settings, including file system ACLs, registry ACLs, service ACLs and group membership, can be managed on many computers at once. For computers that are not joined to an Active Directory domain, the Security Templates and Security Configuration and Analysis MMC snap-in components of the Security Configuration Manager can be used to create security templates and apply them to individual computers.

> **Note**
> Security settings that are defined via domain-based Group Policy always override security settings that are configured locally.

Whistler Professional allows you to configure security settings in the following areas:

- **Account Policies**. This includes password policies such as minimum password length and account lockout parameters.
- **Local Policies**. This includes auditing policy, user rights and privilege assignment, and various security options that can be configured locally on a particular Whistler Professional–based computer.
- **Event Log Settings**. This is used to configure auditing for security events such as successful and failed logon and logoff attempts.
- **Public Key Policies.** These are used to configure encrypted data recovery agents, domain roots, and trusted certificate authorities.
- **Software Restriction Policies:** This is a new Whistler feature that allows you to configure secure operating zones for different applications based on the degree of trust you have in the user and in the application code.
- **IP Security Policy**. This is used to configure network Internet Protocol (IP) security.

- **Restricted Groups**. This is used to manage the members of built-in groups that have certain predefined capabilities. These groups include built-in groups such as Administrators, Power Users, print operators, server operators, and so on, as well as domain groups, such as domain administrators. You can also add groups that you consider to be sensitive or privileged to the Restricted Groups list, along with their membership information. This allows you to track and manage these groups as part of system security configuration or policy.

- **System Services**. This is used to configure and manage security settings for areas such as network services, file and print services, telephony and fax services, and Internet/intranet services. Security policy allows you to configure the service startup mode (automatic, manual, or disabled) as well as security on the service.

- **Registry**. This is used to manage the security descriptors on registry keys.

- **File System.** This is used to configure and manage security settings on the local file system. The configuration file contains a list of fully qualified file or directory paths and security descriptors for each.

Administrators who have implemented an Active Directory domain structure can configure and apply additional security configuration options, such as Kerberos policy, for their Windows 2000 Professional and Whistler Professional clients.

Note

For information about the use and management of Group Policy in Active Directory environments, see "Group Policy Overview" in the *Distributed Services Guide*. For more information about individual policy settings, see Appendix D, "Group Policy Settings that Affect the Desktop" in this book.

Software Restriction Policies

When users run applications, they do so in the security context defined by their rights and restrictions. For example, a user might have the right to view and edit documents using Microsoft Word, but not have the right to install or modify the application itself. These rights and restrictions do not always prevent untrusted applications from taking advantage of the security contexts of trusted applications. The increasing number of "stealth" applications distributed via e-mail and the Internet create a need for a more precise level of

administrative control over the relationship between applications and the user's security context.

Software restriction policies are designed to assist you in regulating unknown or untrusted applications by allowing you to classify applications as trusted or untrusted. After trusted and untrusted applications have been identified, you can then apply a policy that regulates each application's ability to execute. This policy can apply to an entire computer or to individual users.

A software restriction policy consists of a default rule about whether programs are allowed to run and whether there are any exceptions to that rule. The default rule can be set to *Unrestricted* or *Disallowed*. Setting the default rule to *Unrestricted* allows an administrator to define just the set of programs that are forbidden to run. Setting the default rule to *Disallowed* allows administrators to specify only the programs that are known and trusted to run, which is a more secure approach.

How you use software restriction policy depends in large part upon how well you know what software applications your users are running. If you know all of the client software that will be allowed, then you can use software restriction policy to enforce the use of only allowed software. If you do not know all of the applications that your users will run, you can still use restricted software policy in a more limited way by disallowing applications that you are certain you do not want as you learn about them.

The Restricted Software settings include two key nodes: *Levels*, which define the maximum authorization level at which a user is allowed to run a piece of software; and *Rules*, which specify the maximum authorization level at which a piece of software is allowed to run on that computer.

When a piece of software is instantiated for execution, the computer uses the maximum values of these two components to determine the authorization level at which the application is allowed to run. For example, you might decide that Microsoft Word, which is a trusted application from a verifiable source, can run in a Power User's full security context. However, you also might decide that an application that is attached to the Power User's e-mail might not come from a verifiable source. By configuring Restricted Software settings, you can define what constitutes an unrecognized source, and therefore permit it to run only in a User security context — or not all — even though the user is logged on as a Power User. Whistler generates a restricted version of the Power User's access token to provide the appropriate security context for the application.

If, on the other hand, the Administrator has determined that applications of this type must be prohibited from running, and the application requires greater security privileges than its restricted token permits (permission to write to the registry, for example), the application will fail.

Configuring Restricted Software

Restricted Software consists of Levels, which are predefined security contexts; and Rules, which define the file types and file locations (including folders and URLs) of an executable, as well as the identifying data that determines the security level that it is associated with.

Whistler Professional includes the following default Levels for Restricted Software settings:

- **Administrative**. At this level, the application runs in the user's security context.

> **Note**
> Most Windows applications require some Read access to a user's profile and also Write access to the temp folders.

- **Revoked**. Applications at this level are not allowed to run.

> **Note**
> Some executable files with a higher security clearance might attempt to load a DLL with lesser clearances. When this happens, the process will fail and an error message will be generated.

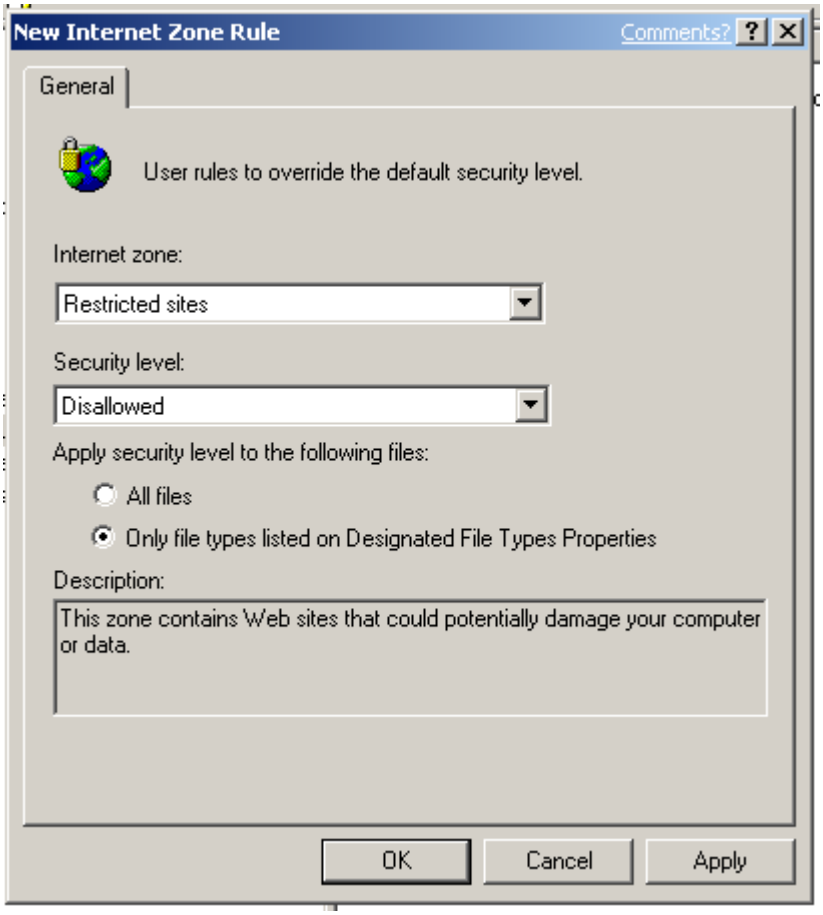
The following Rules can be configured for each Level:

- **Path.** The level of trust based on the file path for an application.
- **Hash.** The level of trust based on the hashed file path for an application.
- **Certificate.** The level of trust based on the certificate associated with an application.
- **Internet Zone.** The level of trust based on the Internet zone (Trusted sites, Restricted sites, local computer, Internet) for an application.

> **Note**
> The Internet Zone Rules apply only to Windows Installer packages.

Figure 18.9 illustrates an Entries dialog box for configuring an Internet zone rule.

Figure 18.9 Restricted Software Entries dialog box



A Path entry can be customized further based on a configurable list of file types. In this way, an Administrator can create a Path Rule that only applies to .EXE and .VBS files.

Security Templates

Security policy settings are combined into a number of preconfigured security templates that have been incorporated into the operating system to address common sets of security

concerns. These templates, which are stored in the *systemroo*t\security\ templates folder, include:

- Setup Security (Setup Security.inf)
- Secure workstation (Securews.inf)
- Highly secure workstation (Hisecws.inf)
- Compatible workstation (Compatws.inf)

Your organization can use these templates as a starting point for customizing your security policy.

Setup Security

The Setup Security template allows you to restore the default security settings. This template is not designed to be deployed through Group Policy. Depending on whether Whistler was installed on a computer as an upgrade or a clean install, as well as what optional components were installed during setup, this template can be different for different computers. Also, you might not want to apply the entire Setup Security template at one time. For example, if you only want to restore the default file system ACLs, then you must specify only the FILESTORE area when applying the template via the **secedit.exe** command-line tool. Otherwise, all other security areas, including registry ACLs, user rights, audit policy, and so on, will be restored to their setup defaults.

Secure

Securews.inf specifies recommended security settings for security elements, such as password policy, audit policy, and so on, that are not likely to have an impact on certified application compatibility. Based on the assumption that Users run applications that are certified for Windows 2000, both the Secure and Highly Secure templates remove all members of the Power Users group.

Highly Secure

Hisecws.inf includes all of the Security template settings and specifies additional security settings which can, depending on the environment, impact functionality. For example, Highly Secure computers will not be able to communicate with computers running versions of Windows earlier than Windows NT 4.0 Service Pack 4, or join domains with domain controllers running versions of the operating system earlier than Windows 2000. The Highly Secure configuration negatively impacts performance and functionality by minimizing the

use of cached data. For some organizations, the security enhancements might not outweigh the functionality or performance tradeoffs.

Based on the assumption that Users run applications that are certified for Windows 2000, both the Secure and Highly Secure templates remove all members of the Power Users group. Both templates must be thoroughly tested in-house before being implemented in production environments.

Compatible

Compatible.inf reduces the security of the Users group in a manner consistent with the requirements of most legacy applications. It is provided for customers that cannot deploy certified applications but do not want their end-users to be Power Users.

Caution

The compatible configuration is not considered to be a secure configuration.

Working with Local Security Policy

To view and implement security policy on a local computer, you need to have Administrator rights to the computer. Unlike Active Directory–based security policy, the local security policy settings on each computer must be configured individually. You can, however, make a security configuration part of a standard desktop configuration that is implemented during the unattended desktop installation process.

Note

For more information about unattended desktop installation, see "Customizing and Automating Installation" in this book.

Administrators can use the following Whistler Professional tools to view and configure some or all of the security policy settings:

- **Security Settings extension to Group Policy**. This Group Policy snap-in replaces the Windows NT 4.0 System Policy Editor and can be used to configure local security policies, as well as security policies for domains or organizational units (OUs). The Group Policy snap-in can be used to configure Account Policies, Local Policies, Public Key Policies, Restricted Software Policies, and IP Security Policies for the local computer.

- **Security Templates snap-in**. This is a stand-alone Microsoft Management Console snap-in that can be used to create a text-based template file that contains security settings for all security areas. The security templates snap-in can be used to configure Account Policies, Local Policies, Event Log settings, Restricted Groups, System Services, Registry settings, and File System settings.

- **Security Configuration and Analysis snap-in**. This is a stand-alone MMC snap-in that can configure or analyze Whistler operating system security. Its operation is based on the contents of a security template that was created using the Security Templates snap-in.

- **Secedit.exe**. This is a command-line version of the Security Configuration and Analysis snap-in. It allows security configuration and analysis to be performed without a graphical user interface (GUI).

Viewing Local Security Policy Settings

You can use the following procedure to view the Security Policy settings on a computer running Whistler Professional.

To view the Security Policy settings using the Group Policy snap-in

- Open the Group Policy snap-in in the Microsoft Management Console and navigate to the **Security Settings** container:

  Local Computer Policy\ Computer Configuration\ Windows Settings\ Security Settings

You can also use the Group Policy snap-in to view the Security Settings container from the command line.

To view the Security Settings container from the command line

1. Open a **Run** command dialog box.

2. At the command line, type:

   ```
   Gpedit.msc
   ```

3. Click **OK.**

Click the + signs that precede every subcontainer under **Security Settings** to view the titles of each individual security setting. For each security setting, the Security Settings extension displays information about any settings that have been enabled.

Modifying Local Security Policy Settings

To modify a local security policy setting, double-click the security item and revise the policy as needed. For example, you can use the following procedure to set the local policy **Prevent users from installing printer drivers**.

To set the local policy to Prevent users from installing printer drivers

1. Click the **+** next to **Local Policies** in the left pane (under Security Settings) to expand it.
2. Click **Security Options**.
3. Double-click **Prevent users from installing printer drivers** in the right pane.
4. Click **Enabled**, and click **OK**.
5. Right-click **Local Policies** (in the left pane), and then click **Reload**.

Reloading the local policy updates the effective policy in the user interface. Depending on the domain or OU password policies that are in effect, the effective policy might or might not have changed on your computer.

Permissions on Group Policy Objects

To edit a Group Policy object, the user must have both Read and Write access to the Group Policy object, and must be one of the following:

- A member of the Administrators group for the local computer, domain, or enterprise.
- A member of the Group Policy Creator Owners group who has previously created the Group Policy object.
- A user with delegated access to the Group Policy object. That is, an administrator or a user who has had access delegated to him or her by someone with the appropriate rights using the **Security** tab on the **Group Policy Object Properties** page.

By default, Group Policy objects allow members of the Domain Administrators, Enterprise Administrators, and Group Policy Creator Owners groups Full Control without the Apply Group Policy attribute set. This means that they can edit the Group Policy object, but the policies contained in that Group Policy object do not apply to them.

By default, Authenticated Users have Read access to the Group Policy object with the Apply Group Policy attribute set. This means that Group Policy affects them.

Domain Administrators and Enterprise Administrators are also members of Authenticated Users; therefore, members of those groups are, by default, affected by Group Policy objects unless you explicitly exclude them.

When a nonadministrator creates a Group Policy object, this person becomes the Creator Owner of the GPO and can edit the GPO. When an administrator creates a Group Policy object, the Domain Administrators group becomes the Creator Owner of the GPO; therefore any member of the Domain Administrators group can edit the GPO.

Creating Group Policy MMC Consoles to Delegate Group Policy

You can delegate Group Policy by creating and saving Group Policy snap-in consoles (.msc files), and then specifying which users and groups have access permissions to the Group Policy object or to an Active Directory container. You can define permissions for a Group Policy object by using the **Security** tab on the **Properties** page of the Group Policy object; these permissions grant or deny access to a Group Policy object to specified groups.

This type of delegation is also enhanced by the policy settings available for MMC. Several policies are available in the **Administrative Templates** node, under **Windows Components** in the **Microsoft Management Console**. These policies enable the administrator to define which MMC snap-ins the affected user might or might not run. The policy definitions can be inclusive, which allows only a specified set of snap-ins to run, or they can be exclusive, which does not allow a specified set of snap-ins to run.

Using Security Templates

The Security Templates snap-in allows you to create a text-based template file that can contain security settings for all of the security areas supported by the Security Configuration Tool Set. You can then use these template files to configure or analyze system security using other tools, in one of the following ways:

- You can import a template file into the Security Settings extension to configure local, domain, or OU security policy.
- You can use the Security Configuration and Analysis snap-in to configure or analyze system security based on a text-based security template.
- You can use the **Secedit.exe** command-line tool directly or in conjunction with other management tools such as Microsoft Systems Management Server or Task Scheduler to deploy a security template or trigger a security analysis.

To load the Security Templates snap-in and view security policy settings

1. Click **Start**, click **Run**, and in the text box, type:

   `MMC /s` (Note: there is a space between the **C** and the **/s**).
2. Click **OK**.
3. Click **Console** (under Console1 in the upper right of the window), click **Add\Remove Snap-in**, and click **Add**.
4. From the list of available Standalone Snap-ins, select **Security Templates**.
5. Click **Add**, then click **Close**.
6. Click **OK**.
7. Click the **+** next to **Security Templates** in the left pane to expand it.
8. Click the **+** next to **C:\WINNT\security\templates** to expand it.

> **Note**
> If you installed Whistler Professional in a different drive or directory, that path will display instead of C:\WINNT.

Creating and Applying Security Templates

You can create your own security template, or you can select the existing template that most closely meets your needs and make any additional changes that you want to that template.

To create a new security template

- Right-click the default templates folder under Security Templates and select **New Template**.

This creates a blank template file without any security settings that you can then customize completely for your organization's needs.

When you have created your template or made all of the appropriate changes to an existing template, it is automatically saved to the templates directory. By using Save As, you can overwrite the existing template of that name or save the new template under a new name.

To save a new or modified security template

- Right-click on the template name, select **Save As** from the context menu, fill in a new name, and click **OK**.

You can save also save your changes by selecting **Save As** from the **Action** menu, typing in the new name, and clicking **OK**.

After you create a security template for your environment, you need to apply it to your computer. When you apply a template to existing security settings, the settings in the template are merged into the computer's security settings.

To apply a security template to a computer

1. Open the **Computer Configuration/Windows Settings** node in the Group Policy snap-in.
2. Right-click and select **Security Settings.**
3. Select **Import Policy** from the secondary menu.
4. Select the template file that you want to import into your environment.
5. Click **OK**.

From the Group Policy snap-in, you can also export the security template for your system.

To export a security template

1. Open the **Computer Configuration/Windows Settings** node in the Group Policy snap-in.

2. Right-click **Security Settings**.

3. Select **Export List** from the secondary menu.

4. Type in the name and location for the Text file that you are exporting.

Alternatively, you can navigate to the \%systemroot%\security\templates folder and copy the appropriate template file to your floppy disk or to another network location.

Performing Security Configuration Tasks Using Security Templates

You can perform the following key security-related tasks using Whistler Professional security templates:

- Configure permissions for a file system directory.
- Create a restricted group policy.
- Inherit, overwrite, and ignore policy changes.

Configuring Permissions for File System Directories

You can use the following procedure to configure permissions for file system directories.

To configure permissions for file system directories

1. Open the Security Templates snap-in and expand one of the templates.

2. Right-click **File System** in the left pane, and click **Add File**.

3. Click the %systemroot**%\repair** directory, and click **OK**. This brings up the Access Control List (ACL) Editor, which allows you to specify permissions for the %systemroot%\repair directory in the Securews.inf template.

4. Select the **Everyone** group in the top pane and click the **Remove** button.

5. Click the **Add** button and select the **Administrators** group. Click **Add** and then click **OK**.

6. Click the **Full Control** checkbox in the bottom pane to give the Administrators group full control permissions.

7. Click the **Advanced** button. Clear the **Inherit from parent the permission entries that apply to child objects** checkbox.

8. Click **OK** twice to accept the Administrator-only permissions defined for the directory.

9. Select the **Replace existing permission on all subfolders and files with inheritable permissions** button and click **OK**.

Creating a Restricted Group Policy

A restricted group policy allows you to define who can and cannot belong to a specific group. When a template (or policy) that defines a restricted group is applied to a system, the Security Configuration Tool Set adds members to the group and removes members from the group to ensure that the actual group membership coincides with the settings defined in the template (or policy). The following procedure describes how to create a restricted group policy.

To create a restricted group policy

1. In the left pane, select one of the security templates, right-click **Restricted Groups**, and select **Add Group**.

2. Type the group name and click **OK**.

3. Double-click the name of the group that you entered in the right pane. You can now define who can be a member of the restricted group and specify other groups that group can be a member of.

4. Click **Add** and then click **Browse**. The **Select Users or Groups** dialog appears.

5. Select the desired user in the **Select Users or Groups** dialog box. Click **Add**.

6. Click **OK** three times.

The restricted group policy defines which users can be members of the specified local group when a specified template is used to configure a Windows 2000 and Whistler system. During configuration, the tool set removes all other users that belong to the group at the time of configuration. Similarly, if, at the time of configuration, the specified user does not belong to the specified group, the Security Configuration Tool Set adds the user to the group.

If no users are specified as members of a defined restricted group (the top box is empty), the Security Configuration Tool Set removes all current members of that group when the template is used to configure a system.

If no groups are specified for a restricted group to belong to (the bottom box is empty), no action is taken to adjust membership in other groups.

Inheriting, Overwriting, and Ignoring Policy Changes

After you define permissions for a file system or registry object, the Security Configuration Tool Set asks you how to configure the object's children.

If you select **Propagate inheritable permissions to all subfolders and files**, normal Whistler ACL inheritance procedures will be in effect — that is, any inherited permissions on child objects are adjusted according to the new permissions defined for this parent. Any explicit access control entry (ACE) defined for a child object remains unchanged.

If you select **Replace existing permissions on all subfolders and files with inheritable permissions**, all explicit ACEs for all child objects (which are not otherwise listed in the template) are removed, and all child objects are set to inherit the inheritable permissions defined for this parent.

To prevent a child object from being overwritten by a parent, the child object can be added to the template and ignored. If a child object is added to the template and ignored, that child's inheritance mode and that child's explicit ACEs remain untouched.

Choosing the option **Do not allow permissions on this file or folder to be replaced** for an object in a template makes sense only if an ancestor of that object is configured to overwrite children. If no ancestor exists in the template, ignoring an object has no impact. If an ancestor exists but is configured so that children inherit, then ignoring a child has no impact.

> By saving a security template file, you can copy your desired configuration settings to multiple computers running Whistler Professional.

You can analyze, summarize, and evaluate your security policy configuration using the Security Configuration and Analysis tools. For more information about using the Security Configuration and Analysis tools, see "Auditing and Analyzing Access Control" later in this chapter.

Auditing and Evaluating Access Control

On many computers, the state of the operating system changes frequently. If you, other administrators, or users periodically make changes to computer configurations, auditing and regular analysis will enable you to validate the security configuration on each computer and to verify that security has not been breached. For example, you might want to track who or what is attempting to perform certain tasks, or you might want to obtain information about why certain events are taking place or not taking place.

Whistler Professional provides a number of auditing and analysis features — including audit policies, the Event Viewer, and the Security Configuration and Analysis Tool Set — that can aid you in effectively validating the security configurations on the computers in your organization.

Enabling Auditing Policies

You can monitor many different types of events on a Whistler Professional system, including user actions such as logons and logoffs, and the success and failure of key application events. Administrators need to monitor these events to track security, system performance, and application errors.

You can set up audit policies to track authorized and unauthorized access to resources. By default, auditing is not enabled. Before you enable auditing, it will be important for you to define exactly what needs to be audited and why you want it to be audited. Auditing can slow down system performance, and it will also require effort on your part to evaluate audit logs; therefore, advanced planning can ensure that you track the appropriate system events without creating excess administrative overhead.

For example, if you decide to audit account logon sessions, you need to consider what the information will be used for. Your security administrators group might be interested in logging failed logon events because this can indicate that someone is trying to log on with an account for which he or she does not have the correct password. Alternatively, you might want to log successful logon attempts to determine whether users are accessing workstations in areas of the network that they are not permitted to use.

To enable auditing, use the Microsoft Management Console with the Group Policy snap-in focused on the local computer. To see the different types of objects for which auditing can be configured, navigate to the following node:

Computer Configuration/Windows Settings/Security Settings/Local Policies/Audit Policy

Here you will find a number of configuration audit policies, which can be used to audit events that fall into the following categories:

- **Account logon events**. Logs an event each time a user attempts to log on. For example, specific events logged include: logon failures for unknown user accounts; time restriction violations; user account has expired; user does not have the right to log on locally; account password has expired; account is locked out. Successful logons also can be tracked through events.
- **Account management**. Logs an event each time an account is managed. This is a useful function if you are concerned about changes being made to user accounts in your environment.
- **Logon events**. Logs an event for logon events that are occurring over the network or generated by service startup (such as an interactive logon or service starting).
- **Object access**. Logs an event each time a user attempts to access a resource such as a printer or shared folder.
- **Policy changes**. Logs an event each time a policy is successfully or unsuccessfully changed in your environment.
- **Use of privileges**. Logs an event each time a user attempts, successfully or unsuccessfully, to use special privileges, such as changing system time.
- **Process tracking**. Logs an event for each program or process that a user launches while accessing a system. Administrators can use this information to track the details of a user's activities while he or she is accessing a system.
- **System events**. Logs designated system events, such as when a user restarts or shuts down a computer.

For each of these categories, determine whether you want to log the Success, Failure, or both Success and Failure for the events they represent. Then configure the object that you want to monitor so that the policy and the object are linked.

For example, to audit file and folder access, you would first mark the activities (Success, Failure) that you want to track under Object Access. Then, for each of the files and folders that you want audited, you would configure the SACLs that enable auditing.

To enable auditing

1. Right-click the file or folder you want audited, and select **Properties**.
2. Select the **Security** tab.
3. At the bottom of the property sheet, select the **Advanced** button.
4. On the **Access Control Settings** page, select the **Auditing** tab.
5. Click the **Add** button, then select one or more Computers, Users, or Groups whose activity you want to monitor.
6. For each entry, determine whether you want to track successes or failures or both. (The list of configurable entries is almost identical to the list of Access Control Entries for files and folders. For more information about ACEs for files and folders, see "Access Control Entries" earlier in this chapter.)
7. Determine whether auditing must be configured on this object only. For example, if the object is a folder and you want to audit activity on files and subfolders, select **Apply these auditing entries to objects and/or containers within this container only**.
8. After you have configured the appropriate settings for all of the users, computers, and groups whose activities you want to track, click **OK** three times to complete the process.
9. In the Group Policy snap-in, navigate to:

   Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Local Policies\Audit Policy
10. Double-click **Audit object access**.

11. Check the appropriate boxes to indicate whether you want **Success** or **Failure** events or both to be logged.

12. Click **OK**.

Auditing the Use of Privileges

Whistler Professional provides the option to audit the use of user privileges. While this setting can be either enabled or disabled, it cannot be applied selectively to individual rights. Auditing the use of user rights generates a very large number of audits, and in most cases the information this audit provides does not outweigh the management costs involved.

Enabling the Use Of Privileges category in the system's Audit policy does not enable the audit of all user rights. The following user rights are never audited:

- Bypass Traverse Checking (SeChangeNotifyPrivilege)
- Generate Security Audits (SeAuditPrivilege)
- Create A Token Object (SeCreateTokenPrivilege)
- Debug Programs (SeDebugPrivilege)
- Replace A Process Level Token (SeAssignPrimaryTokenPrivilege)

The following user rights are audited only if a specific registry setting is present:

- Backup Files and Directories (SeBackupPrivilege)
- Restore Files and Directories (SeRestorePrivilege)

Caution

Do not edit the registry directly unless you have no alternative. The registry editors bypass standard safeguards, allowing settings that can degrade performance, damage your system, or even require you to reinstall Windows. You can safely alter most registry settings by using the programs in Control Panel or Microsoft Management Console. If you must edit the registry directly, back it up first. Read the Registry Editor Help for more information.

To enable auditing of backup and restore privileges

1. At the command line, type:

   ```
   Regedit
   ```

2. Navigate to the following registry key:

   HKLM\SYSTEM\CurrentControlSet\Control\Lsa\FullPrivilegeAuditing

3. Set the value to 1 to enable auditing.

The auditing of backup and restore privileges can also be enabled through the security policy user interface in Whistler Professional.

Tip

Do not audit the use of user privileges unless it is strictly necessary for your environment. If you must audit the use of user privileges, it might be worthwhile to obtain or write an event-analysis tool that can filter only on the user rights that are of interest to you.

Auditing Account Management

The Account Management audit policy is very detailed in Windows 2000 and Whistler and in later service packs of Windows NT 4.0. Enabling auditing for this event category allows you to record the success or failure of the following domain and local events (which are listed with their event numbers):

- 624 User Account Created
- 625 User Account Type Change
- 626 User Account Enabled
- 627 Change Password Attempt
- 628 User Account Password Set
- 629 User Account Disabled
- 630 User Account Deleted
- 631 Security Enabled Global Group Created

- 632 Security Enabled Global Group Member Added
- 633 Security Enabled Global Group Member Removed
- 634 Security Enabled Global Group Deleted
- 635 Security Enabled Local Group Created
- 636 Security Enabled Local Group Member Added
- 637 Security Enabled Local Group Member Removed
- 638 Security Enabled Local Group Deleted
- 639 Security Enabled Local Group Changed
- 640 General Account Database Change
- 641 Security Enabled Global Group Changed
- 642 User Account Changed
- 643 Domain Policy Changed
- 644 User Account Locked Out
- 645 Computer Account Created
- 646 Computer Account Changed
- 647 Computer Account Deleted
- 648 Security Disabled Local Group Created
- 649 Security Disabled Local Group Changed
- 650 Security Disabled Local Group Member Added
- 651 Security Disabled Local Group Member Removed
- 652 Security Disabled Local Group Deleted
- 653 Security Disabled Global Group Created
- 654 Security Disabled Global Group Changed

- 655 Security Disabled Global Group Member Added
- 656 Security Disabled Global Group Member Removed
- 657 Security Disabled Global Group Deleted
- 658 Security Enabled Universal Group Created
- 659 Security Enabled Universal Group Changed
- 660 Security Enabled Universal Group Member Added
- 661 Security Enabled Universal Group Member Removed
- 662 Security Enabled Universal Group Deleted
- 663 Security Disabled Universal Group Created
- 664 Security Disabled Universal Group Changed
- 665 Security Disabled Universal Group Member Added
- 666 Security Disabled Universal Group Member Removed
- 667 Security Disabled Universal Group Deleted
- 668 Group Type Changed
- 669 Add SID History (Success)
- 670 Add SID History (Failure)

Using the Event Viewer

The Event Viewer is an MMC snap-in that enables you to view three different logs that are stored by Windows 2000 and Whistler:

- **System log.** The system log contains events logged by the Windows 2000 system components, such as drivers or other system components that failed to load during startup. Windows 2000 predetermines the event types logged by the system components.

- **Application log.** The application log contains events logged by applications or programs. For example, a database program might record a file error in the application log. The program developer decides which events to record. Many Windows 2000 services (such as DHCP, DNS, File Replication Services, and so forth) use the application log.

- **Security log.** The security log, if configured to do so, records security events, such as valid and invalid logon attempts. Events that are related to resource use, such as creating, opening, or deleting files, can also be logged. An administrator can specify the events that are recorded in the security log policy.

When you select the log type in the left pane of the MMC, the corresponding log data is displayed in the right pane.

The data in the right pane can be filtered and resorted. In addition, you can select which columns of data to present by selecting **Choose Columns** from the **View** menu.

The Event Viewer tracks five basic types of events:

- **Error.** A significant problem, such as loss of data or loss of functionality.

- **Warning.** An event that is not necessarily significant but might indicate a possible future problem.

- **Information.** An event that describes the successful operation of an application, driver, or service.

- **Success Audit.** An audited security event in which a user's attempt to access a resource succeeds.

- **Failed Audit.** An audited security event in which a user's attempt to access a resource fails.

In addition, the Event Viewer records the following:

- The date of the event.

- The time at which the event occurred.

- The source (such as a service or process) that reported the event to the Event Viewer.
- The category of the event. In many cases, the category relates to the subsystem that reported the event.
- The user account associated with the event.
- The computer on which the event occurred.
- The Event ID, which is a numeric code that can be used to obtain additional information regarding the event being logged.
- A description of the event.

Using the Security Configuration and Analysis Tool Set

You can use the Security Configuration and Analysis Tool Set to analyze current system settings against a baseline template at any time. Performing this analysis allows you to do the following:

- Identify security holes that might exist in a current configuration.
- Identify the changes that a potential security policy will transmit to a system before actually deploying the security policy.
- Identify deviations from a policy that is currently imposed on a system.

For example, if you have created a custom security template, the Security Configuration and Analysis tools will allow you to compare your system's current settings against the settings that are defined by the security template that you created. If the custom security template defines a more secure configuration than the current settings provide, the analysis will identify the security holes that exist in the current system configuration, as well as the changes that will take place if the custom template is used to configure the system.

To load the Security Configuration and Analysis MMC snap-in:

1. On the **Start** menu, click **Run** and type: **MMC /s**

2. From the **Console** menu, select **Add\Remove Snap-in**, and click **Add**.
3. Select **Security Configuration and Analysis**.
4. Click **Add,** click **Close**, and then click **OK**.

Creating and Analyzing a Security Configuration Database

All configurations and analyses are database-driven. The security configuration and analysis database, which is also referred to as the local computer policy database, is a computer-specific data store that is generated when one or more configurations are imported to a particular computer. A security configuration and analysis database is the starting point for all configurations and analyses done on a system.

An initial database is created during a clean install of Windows 2000 or Whistler Professional. Initially, it contains the default security configurations that are provided with the system. You can export and save this database to a security configuration file immediately after the installation, for use in the event that you want to restore the initial security configuration at a later point.

This database defines the security policy that is in force for that system. At any time, the system runs with the configuration defined in security policy. However, security policy might not define the entire configuration. For example, security for every file or folder path might not be defined. This implies that security configuration attributes that are not enforced by policy can take any value — either a default value or a value defined by another mechanism, such as the ACL Editor in Windows Explorer — for file and folder security. Attributes that are not enforced by policy might also be configured manually using personal databases. However, any custom configurations that conflict with the policy are overridden by the definitions in the policy. Personal database configurations are useful in areas like the registry and the file system, where multiple users on the system can secure their own registry hive and home directory subtrees.

You can use the Security Configuration Manager to compare the current system configuration against the stored configuration in the database. Performing this analysis provides you with information about where a particular system deviates from the stored configuration. This information is useful for troubleshooting problems, tuning the security policy, and, most importantly, detecting any security flaws that might open up in the system over time.

The database is initially created from the computer-independent configuration file described above. New configurations can be added to the database incrementally without overwriting the entire configuration.

To generate a security configuration database

1. Select and right-click **Security Configuration and Analysis** in the left pane.
2. Click **Open Database**.
3. Type a name for your new database.
4. Click **Open**.
5. Select an existing security template to import into the database.
6. Click **Open**.

The name of the database will appear in the result pane. Several more options are available when you right-click **Security Configuration and Analysis**.

To analyze the security configuration database

1. Right-click **Security Configuration and Analysis**.
2. Select **Analyze Computer Now** from the context menu.
3. Specify a log file for the analysis results, such as the following:

   ```
   %windir%\security\logs\Mysecurews.log
   ```
4. Click **Open,** and then click **OK**. A progress dialog displays as the analysis proceeds.

Reviewing the Results of a Database Analysis

After a database analysis has been completed, you can view the security information under the Security Configuration and Analysis node.

To view the results of a database analysis

1. From the **Security Configuration and Analysis** node, click **View**.
2. Select the **Description Bar** to expose the database you are currently working with.

3. Expand **Security Configuration and Analysis** in the left pane, then expand **Local Policies**, and click **Security Options**.

You can double-click any setting in the result pane to further investigate discrepancies and modify database settings if desired.

Configuration results are displayed for the following areas:

- **Account policies**. This includes password, account lockout, and Kerberos policies. Kerberos policies are relevant only on Windows 2000 or Whistler domain controllers.
- **Local policies**. This includes audit policy, user rights assignment, and computer security options.
- **Restricted groups**. This includes group memberships for selected groups that you consider to be sensitive.
- **Object trees**. This includes directory objects (in Windows 2000 and Whistler domain controllers), registry keys, and the local file system.
- **System services**. This includes all local or network system services.

> **Note**
> For each object tree, defined configuration files allow you to configure (and analyze) settings for security descriptors, including object ownership, the Access Control List (ACL), and auditing information.

In the right pane, both database and actual system settings are displayed for each object. Discrepancies are highlighted with a red flag. Consistencies are highlighted with a green check mark. If neither a flag nor a check mark appears, the security setting is not specified in the database (that is, the security setting was not configured in the template that was imported).

Modifying Baseline Analysis Settings

After you review the analysis results, you might decide to update the baseline database that was used to perform the analysis, if you find that you have changed your mind about the relevancy of the security specification that was originally defined for an object.

If you consider an object to be relevant to security, check the **Define this policy in the database** checkbox when viewing the detailed analysis results. If this box is unchecked, the object is removed from the configuration and receives its inheritance from the parent object, as defined.

If you want to base future configurations or analyses on a different security specification, you can click the **Edit Security Settings** control to modify the security definition currently stored in the database.

Configuring and Analyzing Operations Using Secedit.Exe

The configuration and analysis operations available from the Security Configuration and Analysis user interface can also be performed using the **Secedit.exe** command-line tool. Using the command-line tool allows you to perform security configuration and analysis in conjunction with other administrative tools, such as Microsoft Systems Management Server or the Task Scheduler built into Windows 2000. **Secedit.exe** also provides some capabilities that are not available in the graphical user interface.

The **Secedit.exe** command allows the following high-level operations:

- **Analyze**. This corresponds to the same task available using the Security Configuration and Analysis snap-in.
- **Configure**. This corresponds to the same task available using the Security Configuration and Analysis snap-in.
- **Export**. This dumps database configuration information into a template (.inf) file. This feature is also available in the snap-in through the Security Configuration and Analysis context menu after a database has been opened.
- **Validate**. This verifies the syntax of a template created using the Security Templates snap-in.

> **Note**
> For information on command line syntax for **Secedit.exe**, see "Automating security configuration tasks" in Whistler Professional Help.

All **Secedit.exe** configurations and analyses are database driven. Therefore, **Secedit.exe** supports parameters for specifying a database (/db) as well as a configuration file (/cfg) to be imported into the database before performing the configuration.

By default, the configuration file is appended to the database. To overwrite existing configuration information in the database, use the /overwrite switch. As with the snap-in, you can specify a log file (/log). **Secedit.exe** also allows detailed (/verbose) log information to be recorded.

Note

While the snap-in always configures all security areas, **Secedit.exe** allows you to specify areas (/areas) to be configured. Security areas not specified with the /areas switch are ignored even if the database contains security settings for those areas.

Additional Resources

Related Information in the Resource Kits

- "Logon and Authentication" in this book for more information about the authentication process and how security contexts are created.
- "Authorization and Access Control" in the *Distributed Services Guide* for more information about authorization in Active Directory environments.

Beta Disclaimer

Back To Top

This documentation is an early release of the final documentation, which may be changed substantially prior to final commercial release, and is confidential and proprietary information of Microsoft Corporation. It is disclosed pursuant to a non-disclosure agreement between the recipient and Microsoft. This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results of the use of this document remains with the user. Unless otherwise noted, the example companies, organizations, products, people and events depicted herein are fictitious and no association with any real company, organization, product, person or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Active Accessibility, Active Channel, Active Client, Active Desktop, Active Directory, ActiveMovie, ActiveX, Authenticode, BackOffice, Direct3D, DirectAnimation, DirectDraw, DirectInput, DirectMusic, DirectPlay, DirectShow, DirectSound, DirectX, DoubleSpace, DriveSpace, FrontPage, IntelliMirror, IntelliMouse, IntelliSense, JScript, Links, Microsoft, Microsoft Press, Microsoft QuickBasic, MSDN, MS-DOS, MSN, Natural, NetMeeting, NetShow, OpenType, Outlook, PowerPoint, SideWinder, Slate, TrueImage, Verdana, Visual