eTrust® Access Control for Windows

Administrator Guide

r8 SP1

CA Product References

This document references the following CA products:

- eTrust® Access Control (eTrust AC)
- eTrust® Single Sign-On (eTrust SSO)
- eTrust® Web Access Control (eTrust Web AC)
- eTrust® CA-Top Secret®
- eTrust® CA-ACF2®
- eTrust® Audit
- Unicenter® TNG
- Unicenter® Network and Systems Management (Unicenter NSM)
- Unicenter® Software Delivery

Contact Technical Support

For online technical assistance and a complete list of locations, primary service hours, and telephone numbers, contact Technical Support at http://ca.com/support.

Contents

Chapter 1: Introduction

This section contains the following topics:

About this Guide

This guide describes the concepts used by eTrust AC for Windows-a product that provides a total security solution for open systems. The guide describes eTrust AC, especially the user interface used to administer eTrust AC for Windows, Policy Manager.

Who Should Use this Guide

This guide was written for security and system administrators who are implementing and maintaining an eTrust AC-protected environment.

Command Notation Conventions

The eTrust AC documentation uses a few special conventions when explaining command syntax and user input:

Format	Meaning
`Mono-spaced font`	Code or program output
Italic	Placeholder for information that you must supply
Bold	Elements that you must type exactly as shown
Between square brackets ([])	Optional items
Between braces ({}); choices separated by pipe (|).	Set of mandatory choices from which you must choose only one
Space and a backslash at end of line (\)	Command continues on the following line

Notes:

- Bold text is also used for simple emphasis. For example:

  You should **never** tape your password to the monitor.

- Sometimes a command does not fit on a single line in this guide. In these cases, a space followed by a backslash (\) at the end of a line indicates that the command continues on the following line.

  Note: Avoid copying the backslash character as it is not needed in the actual command syntax.

- A pipe (|) separates mutually exclusive items. The set of items is enclosed in braces ({}), which you are **not** intended to type when you type one of the items. For example, the following means **either** a user name **or** a group name:

  {*username*|*groupname*}

Example: command notation conventions

The following code illustrates how command conventions are used in this guide:

ruler *className* [**props**({**all**|{*propertyName1*[,*propertyName2*]})]

In this example:

- The command name (ruler) is shown in bold as it must be typed as shown.
- The *className* option is in italic as it is a placeholder for a class name (for example, USER).
- You can run the command without the second part enclosed in square brackets, which is optional.
- When using the optional parameter (props), you can choose the keyword *all* or, specify one or more property names separated by a comma.

Chapter 2: Basic Concepts

This section contains the following topics:

eTrust AC

eTrust AC is a software product that is an active, comprehensive security software solution for Open Systems, tied dynamically to the operating system. Each time a user requests a security-sensitive operation-such as opening a file, substituting a user ID, or obtaining a network service-eTrust AC can intercept the event in real time and evaluate its validity before passing control to the standard operating system (OS) functions.

What Is Access Control?

eTrust AC provides you with a powerful tool for managing security for your native platforms, making it possible to implement a security policy that can be customized entirely to an enterprise's security requirements. eTrust AC lets you provide security for users, groups, and resources beyond what is available in native operating systems, to centrally manage security across the organization, and to integrate your Windows and UNIX security policies in a heterogeneous environment.

What Is Protected?

eTrust AC protects the following entities:

- **Files**

  Is a user authorized to access a particular file?

  eTrust AC restricts a user's ability to access a file. You can give a user one or more types of access, such as READ, WRITE, EXECUTE, DELETE, and RENAME. The access can be specified regarding an individual file or to a set of similarly named files.

- **Terminals**

  Is a user authorized to use a particular terminal?

  This check is done during the login process. Individual terminals and groups of terminals can be defined in the eTrust AC database, with access rules that state which users, or groups of users, are allowed to use the terminal or terminal group. Terminal protection ensures that no unauthorized terminal or station can be used to log into the accounts of powerfully authorized users.

- **Signon time**

  Is a user authorized to log on at a particular time on a particular day?

  Most users use their stations only on weekdays and only during work hours; the time-of-day and day-of-week login restrictions, as well as holiday restrictions, provide protection from hackers and from other unauthorized accessors.

- **TCP/IP**

  Is another station authorized to receive TCP/IP services from the local computer? Is another station authorized to supply TCP/IP services to the local computer? Is another station permitted to receive services from every user of the local station?

  The advantage of an open system-a system in which both the computers and the networks are open-is also a disadvantage. Once a computer is connected to the outside world, one can never be sure who enters the system and what damage an alien user may do, whether intentionally or by mistake. eTrust AC includes “firewalls” that prevent local stations and servers from providing services to unknown stations.

- **Multiple login privileges**

  Is the user permitted to log in from a second terminal?

  The term *concurrent logins* refers to a user's ability to be logged onto the system from more than one terminal. eTrust AC can prevent a user from logging in more than once. This prevents intruders from logging into the accounts of users who are already logged in.

- **User-defined entities**

  You can define and protect both regular entities (such as TCP/IP services and terminals) and functional entities (known as *abstract* objects; such as performing a transaction and accessing a record in a database). The Application Programmer's Interface (API) that is used to define and protect abstract objects is described in the *SDK Developer Guide*.

- **Aspects of administrator authority**

  eTrust AC provides the means to both delegate administrator authorities to operators and restrict root itself.

  eTrust AC provides the means to delegate administrator authorities to operators and to restrict the administrator itself.

- **Registry keys**

  Is a user authorized to access a particular registry key?

  eTrust AC restricts a user's ability to access registry keys. You can give a user one or more types of access, such as READ, WRITE, and DELETE. The access can be specified with regard to an individual registry key or to a set of similarly named registry keys.

- **Programs**

  Can a particular program be trusted? Is the user authorized to invoke it? Can the user access a specific resource using a program?

  The security administrator can test programs to ensure that they do not contain any security loopholes that can be used to gain unauthorized access. Programs that pass the test and are considered safe, are defined as trusted programs. The eTrust AC self-protection module (also referred to as the **watchdog**) knows which program is in control at a particular time and checks whether the program has been modified or moved since it was classified as trusted. If a trusted program is modified or moved, the program is no longer considered trusted and eTrust AC does not allow it to run.

In addition, eTrust AC protects against various deliberate and accidental threats, including:

- **Kill attempts**

  eTrust AC can be used to protect critical servers and services or daenins against kill attempts.

- **Password Attack**

  eTrust AC protects against various types of password attacks, enforces the password-definition policies of your site, and detects break-in attempts.

- **Password Delinquency**

  eTrust AC policies delineate rules that force users to create and use passwords of sufficient quality. To ensure that users create and use acceptable passwords, eTrust AC can set maximum and minimum lifetimes for passwords, restrict certain words, prohibit repetitive characters, and enforce other restrictions. Passwords are not permitted to last too long.

- **Account Management**

  eTrust AC policies ensure that dormant accounts are dealt with appropriately.

How Is It Protected?

eTrust AC services start immediately after the operating system finishes its initialization. eTrust AC places hooks in system services that must be protected. In this way, control is passed to eTrust AC before the service is performed. eTrust AC decides whether the service should be granted to the user.

For example, a user may attempt to access a resource protected by eTrust AC. This access request generates a system call to the kernel to open the resource. eTrust AC intercepts that system call and decides whether to grant access. If permission is granted, eTrust AC passes control to the regular system service; if eTrust AC denies permission, it returns the standard permission-denied error code to the program that activated the system call, and the system call ends.

The decision is based on access rules and policies that are defined in the database. The security administrator defines most of the records in the database.

The database describes two types of objects: accessors and resources. *Accessors* are users and groups. *Resources* are objects to be protected, such as files and services. Each record in the database describes an accessor or a resource.

Each object belongs to a class-a collection of objects of the same type. For example, TERMINAL is a class containing objects that are terminals (workstations) protected by eTrust AC.

Class Activation

The information about CLASS status (that is, whether the class was active or inactive) is held in the database. Every attempt to access a resource is intercepted by eTrust AC, which checks the status in the database. If the class is inactive, access is allowed without further checking for authorization.

eTrust AC issues a list of active classes when the Engine starts up and when a user changes the CLASS activity status. If a class is inactive, access to the resource is not intercepted, reducing overhead.

Accessor Elements

Each user is represented by an *accessor element* (ACEE)-an in-memory reflection of the user's record in the database. eTrust AC builds the accessor element during the login process. The accessor element is associated with the user's process. Whenever the process requests a system service that is protected by eTrust AC, or issues an implicit request to access a resource, eTrust AC accesses the resource's record. It then determines whether the information in the previously created accessor element-such as the user's security level, mode, and group-lets the user access the resource.

Components

eTrust AC includes a database (seosdb), two drivers (seosdrv and drveng), a number of services (including the Watchdog, the Agent, the Engine (seosd), the Policy Model, and Task Delegation), and a graphical user interface.

Database

The database contains definitions of the following elements:

- Users and groups in your organization
- System resources that need protection
- Rules governing user and group access to system resources

Drivers

The drivers protect all the eTrust AC files and registry keys by performing the following tasks:

- Intercepting every request to open a file or registry key, terminate a process, and perform network activities
- Passing these requests to the eTrust AC Engine and receiving the decision of the Engine whether the request should be granted or denied
- Forwarding the decision to the original system call of the operating system, which then continues its processing based on the answer it received from the drivers.

Services

Watchdog

The Watchdog constantly checks that the other eTrust AC services are running. On the rare occasion when the Watchdog discovers that another service has stopped, it immediately starts the service again.

Agent

The Agent is responsible for the following tasks:

- Communicating with eTrust AC clients through a proprietary application protocol above TCP/IP
- Managing security for the eTrust AC user

Engine

The Engine is responsible for the following tasks:

- Managing the database, including controlling all database updates
- Deciding whether to grant access requests that it receives from the Driver and the Agent
- Checking that the Watchdog service is running, and restarting the Watchdog if it discovers that the Watchdog has stopped running

The Engine handles database access requests **and** makes the access decision, creating an efficient service.

Policy Model

Managing tens or hundreds of databases individually is not practical. Therefore, eTrust AC supplies the Policy Model service, a component that permits management of many computers from one computer. Using the Policy Model service is optional, but it greatly simplifies administration at large sites.

With the Policy Model service, use a Policy Model database (PMDB). Like other eTrust AC databases, the PMDB contains users, groups, protected resources, and rules governing access to the resources. In addition, the PMDB contains a list of subscriber stations. A subscriber station is one linked to the PMDB so that any change to the PMDB is automatically sent to the subscriber database.

You can create a basic security policy for your organization and implement all the necessary rules on a single database-the Policy Model database. The subscribers can include both Windows and UNIX stations, ensuring uniform rules with minimal administrative effort.

The system or security administrator updates the PMDB. The PMDB then propagates all updates from the PMDB to its subscribers in batch mode, freeing the administrator for other work.

A PMDB can have two types of subscribers: another PMDB or a local database. This PMDB also contains a list of subscribers to which it propagates database updates. This feature lets you build a hierarchy of PMDBs. The local database can be used to protect the users, groups, and resources defined on the station.

Graphical User Interface

Policy Manager (see page 39) is the graphical user interface through which all eTrust AC functions are carried out.

Features

eTrust AC lets you manage native Windows from one central location and significantly extends native Windows security. eTrust AC can also protect itself. These features are described in the following sections.

Managing Windows

Once eTrust AC is installed on the Windows stations in your organization, you can manage all of them, regardless of the domains they are in, from one central station. You do this by using the Policy Manager interface or the command-line language called selang.

Providing Self-Defense

It is virtually impossible for hackers or users to bring down eTrust AC services, intentionally or unintentionally. When eTrust AC is running, it is also virtually impossible for unauthorized users to change or erase eTrust AC files and data.

Administering Native Windows Security

The following elements of Windows security can be administered with eTrust AC.

Registry Protection

The Windows registry is a centralized database that contains most of the operating system parameters, including the parameters that control device drivers, configuration details, and hardware, environment, and security settings.

eTrust AC protects the registry to ensure that an unauthorized user does not change system parameters. Authorized users can update registry settings as necessary.

Active Directory

Active Directory is the directory service used in Windows operating systems since Windows 2000; it provides a hierarchically structured repository of information about objects in the network, such as users, computers, and services.

When you have Active Directory installed on your Windows operating system, you can use eTrust AC to add and modify users and groups and the extended user and group properties, just as you can administer users and groups in the native Windows environment.

When you have Active Directory installed on your Windows server, you can use the OU class to create users, groups, and computers within a specific organizational unit.

File Protection

Windows supports the use of several different types of file systems. The most common are FAT and NTFS. If you use the NTFS file system, Windows protects the files in your system by creating and updating ACLs for each file. eTrust AC supports file ACLs.

Password Protection

Native Windows security can protect passwords and enforce password quality in a number of ways. Windows offers the ability to:

- Enforce a maximum password age
- Enforce a minimum password length
- Save up to 24 generations of a user's passwords
- Lock out accounts after repeated login failures
- Force users to log on to Windows before changing their passwords

eTrust AC also enforces the same rules but through its own unique mechanisms. In addition, eTrust AC implements two-way password synchronization with mainframe computers.

Expanding Native Windows Security

The following eTrust AC features expand native Windows security.

Administrator Account Limitations

Users who administer and manage Windows are usually members of predefined groups that are automatically created during system setup. Each of the predefined groups exists to perform a certain set of system functions. Users that are members of a group are permitted to perform all the functions of the group.

The most powerful group in Windows is the Administrators group. Every member of the Administrators group can perform a wide range of tasks, from creating, deleting, and modifying users to locking, reconfiguring, and shutting down servers.

One of the major security risks in Windows is that an unauthorized user can gain control of a user account in the Administrators group. If this happens, the unauthorized user can cause enormous damage to the system.

eTrust AC lets you limit the rights granted to the Administrator account and to limit the rights of users who are members of the Administrators group. This reduces the vulnerability of your Windows system.

Granting Administrator Rights to Regular Users

eTrust AC lets you grant ordinary users (that is, non-administrators) the necessary rights and privileges so that these users can perform administrative tasks without being members of the Administrators group. This is called *task delegation*. The ability to delegate tasks-grant administrative privileges-in this granular way is one of the most significant advantages of eTrust AC.

- A record in the SUDO class stores a command script to allow users to run the script with borrowed permissions.
- The data property value is the command script. This value can be modified by adding to it optional script parameter values.
- Each record in the SUDO class identifies a command for which a user can borrow permissions from another user.
- The key of the SUDO class record is the name of the SUDO record. This name is used instead of the command name when a user executes the commands in the SUDO record.

Defining SUDO Records

A record in the SUDO class stores a command script so that users can run the script with borrowed permissions. The ability to borrow permissions is tightly controlled by the SUDO record, as well as by the sesudo command that executes the scripts.

Note: The sesudo command cannot execute interactive processes when the SeOS Task Delegation service runs under a user account other than the SYSTEM account on a machine where Terminal Services are installed.

In a SUDO record, the comment property is used for a special purpose, and often it is known by its alternate name: the data property.

The comment property's value is the command script, with the optional addition of one or more script parameter values that are to be prohibited or permitted. The entire comment property value must be enclosed in single quotes, and executables should be referenced by their complete path names in order to prevent Trojan horses from taking their place.

This is the format for the comment property:

```
comment('cmd[;[prohibited-values][;permitted-values]]')
```

Because the lists of prohibited and permitted values are optional, a simple comment property value can be the following:

```
newres SUDO NET comment('net use')
```

The simple value in the command means that the command sesudo NET will execute the command 'net use'. No particular script parameter values are prohibited; all are permitted.

Wildcards and powerful variables give you flexibility in specifying prohibited and permitted parameters. The wildcards you can use are the standard Windows wildcards. Prohibited and permitted parameters can also contain the following variables:

Variable	Description
$A	An alpha value
$G	An exiting eTrust AC group name
$H	A parameter that starts with the user's home directory
$N	A numeric value
$O	The eTrust AC name of the user running sesudo

Variable	Description
$U	An exiting eTrust AC user name
$e	An empty entry. Use this to specify a SUDO command with no parameters for the rule.
$f	An existing file name
$g	An existing Windows group name
$h	An existing host name
$r	An existing file with Windows read access
$u	An existing Windows user name
$w	An existing file with Windows write access
$x	An existing file with Windows execute access

If you append a list of *prohibited* parameter values to the script:

- Separate the script from the prohibited parameter values with a semicolon, but keep them all inside the single quotes. For example, if you want to prevent the user from using -start but you permit the user to use all other parameters, enter the following command:

  ```
  newres SUDO scriptname comment('cmd;-start')
  ```

  where *cmd* represents your script.

  Alternatively, if you do not allow any parameter values, but rather want all parameters defaulted, define the SUDO record as follows:

  ```
  newres SUDO scriptname comment('cmd;*')
  ```

- If a script parameter has more than one prohibited value, use the space character as a separator. For example, if you want to prevent the user from using -start and -stop but you permit the user to use all other parameters, enter the following command:

  ```
  newres SUDO scriptname comment('cmd;-start -stop')
  ```

- If more than one script parameter has prohibited values, use the pipe character (|) as a separator between sets of prohibited values. For example, if you want to prevent the user from using -start and -stop for the script's first parameter and from using any existing Windows user name for the second parameter (see the previous list of variables), enter the following command:

  ```
  newres SUDO scriptname comment('cmd;-start -stop | $u')
  ```

  If the script has more parameters than you list, then your last set of prohibited parameters applies to all the remaining parameters.

If you append a list of *permitted* parameter values to the script,

- The sesudo utility checks that the parameter values:
  - Do *not* match any of the corresponding *prohibited* values.
  - Match at least one of the corresponding *permitted* values.

  This means that if a parameter value is in the prohibited list, it will not be permitted even if it is also specified in the permitted list.

- Separate the list of *permitted* values from the list of *prohibited* values with a semicolon, but keep them all inside the single quotes. Even if you have no list of prohibited values, you still need the semicolon; otherwise what you intend to permit will be prohibited. For example, if you want to allow only the value NAME as a parameter value for the script, enter the following command:

  ```
  newres SUDO scriptname comment('cmd;;NAME')
  ```

- Just as in the other list,

- If a script parameter has more than one permitted value, use the space character as a separator.
- If more than one script parameter has permitted values, use the pipe character (|) as a separator between sets of permitted values.

For example, if you have two parameters, and the first must be numeric but must not be a Windows user name, and the second must be alphabetic but must not be a Windows group name, enter the following command:

```
newres SUDO scriptname comment('cmd;$u | $g ;$N | $A')
```

If the script has more parameters than you list, then your last set of permitted parameters applies to all the remaining parameters.

Thus, the overall format for the comment property is this: first the script; then the prohibited values, parameter by parameter; then the permitted values, parameter by parameter:

```
comment('cmd; \
param1_prohib1 param1_prohib2 ... param1_prohibN | \
param2_prohib1 param2_prohib2 ... param2_prohibN | \
 ...
paramN_prohib1 paramN_prohib2 ... paramN_prohibN ; \
param1_permit1 param1_permit2 ... param1_permitN | \
param2_permit1 param2_permit2 ... param2_permitN |
 ...
paramN_permit1 paramN_permit2 ... paramN_permitN')
```

The sesudo utility checks each parameter entered by the user in the following manner:

1. Test if parameter N matches permitted parameter N. (If permitted parameter N does not exist, the last permitted parameter is used.)
2. Test if parameter N matches prohibited parameter N. (If prohibited parameter N does not exist, the last prohibited parameter is used.)

If all the parameters match permitted parameters, and none match prohibited parameters, sesudo executes the command.

Task Delegation Examples

Example 1

1. Select the Resources icon  from the Access Control program bar.

   The Resources window displays.

2. Expand the Task Delegation Resource tree, right-click Tasks, and select New.

   The Create New SUDO Resource - General dialog appears, letting you create a new policy.

3. In the Name field, enter the name *NET* for the SUDO record.

   In the Data field, enter the following:

   ```
   net;start;send
   ```

   The format for the data property is as follows:

   ```
   command; prohibited-values; permitted-values
   ```

   For example, to allow the execution of *net send* and prevent the execution of *net start* for user *any_user*, enter the following in the Data filed and authorize any_user to this SUDO record:

   ```
   net;start;send computer_name message
   ```

4. For the Owner field click Browse, select *nobody* from the Users tab, and click OK.

5. Click Set Default Access, select None, and click OK.

6. Select the Authorize icon from the left panel of the dialog.

   The Authorize page of the dialog appears.

7. Click Insert  to add an accessor and then click Browse next to the Name field.

8. Select a user or group to delegate permissions to and click OK.

   The Add/Edit eTrust Accessor dialog box appears.

9. Click OK.

10. Check Execute Permissions and click OK.

    The new SUDO resource is created.

11. Perform a SUDO record test.
    a. Log in as the user to which the SUDO record was applied.
    b. Open the command prompt and execute the following:

       ```
       sesudo -do NET start
       ```

       The following message appears:

       ```
       sesudo: you are not allowed to use 'start' as parameter number 1.
       ```

       Note: *net start* will not execute because it was defined as a prohibited value.
    c. Execute the following value:

       ```
       sesudo -do NET send
       ```

       The command should execute.

Example 2

A user can perform highly privileged operations using any snap-in MSC module, as the following example shows:

1. In the Resources window, expand the Task Delegation Resource tree, right-click Tasks, and select New.

   The Create New SUDO Resource - General dialog appears, letting you create a new policy.
2. In the Name field, type *services*.
3. In the Data field, type *c:\winnt\system32\mmc.exe*
4. In the Owner field, select *nobody*.
5. Select the *Interactive* check box.

   The Interactive feature provides the desktop user interface that can be used by whoever is logged in when the service is started. This is available only if the service is running as a LocalSystem account.
6. Click Set Default Access, select None, and click OK.
7. Select the Authorize icon from the left panel of the dialog.

   The Authorize page of the dialog appears.
8. Click Insert  to add an accessor and then click Browse next to the Name field.
9. Select a user or group to delegate permissions to and click OK.
10. Check Execute Permissions and click OK.

    The new SUDO resource is created.
11. Perform a SUDO resource test.
    a. Log in as the user to which the SUDO resource was applied.

b. Open the command prompt and execute the following:

```
sesudo -do services
```

c. mmc.exe will start.

12. To deny execution of the SUDO resource, edit the SUDO Authorize properties and select the check box in the Deny column.

13. Open the command prompt and enter the following:

```
sesudo -do services
```

The following message appears:

```
sesudo: you are not authorized to use services command.
```

Note: *services* is the name of the SUDO resource created for this example. The SUDO resource should prevent the execution of the services script.

Enhanced File Protection

eTrust AC supports both logical and absolute file name formats. For example, if the file foo.txt is located under the directory \tmp on the logical drive D and the logical name "D:" is assigned to physical disk 1, partition 0, you can use either the logical or absolute file name to define a file to the eTrust AC database:

```
nr file D:\tmp\foo.txt
```

or

```
nr file \Device\HardDisk1\Partition1\tmp\foo.txt
```

Note: If the second format is used, the file remains protected even if the logical name of the disk is changed. The absolute file name format is also supported for eTrust AC generic file protection.

eTrust AC protects all file systems currently used with Windows. The two most commonly used are the Windows file system (NTFS) and the file allocation table (FAT). eTrust AC also supports CDFS (a file system especially for CDs) and HPFS (an OS/2 file system).

eTrust AC supplies a total security solution to the file allocation table (FAT) and an extra layer of security to other file systems including NTFS and CDFS.

Generic File Protection

eTrust AC supports both logical and absolute file names. The absolute file name format is also supported for eTrust AC generic file protection.

Generic file protection lets you protect all the files that fit a specified wildcard pattern (regular expression). Any resource with a name matching the specified wildcard pattern is protected by the specified generic access rule. eTrust AC lets you protect files generically.

Should a resource match more than one generic access rule, eTrust AC chooses the rule that most closely matches the file.

With generic file protection, no more than a handful of security rules must be defined to protect many of the files requiring protection.

Enhanced Password Protection

Native Windows security provides a significant amount of protection for user passwords (see page 19). However, eTrust AC significantly extends password protection so that the likelihood of a hacker succeeding in stealing a password is greatly reduced.

When using eTrust AC, you can create additional rules that force users to choose safer, more secure passwords. For instance, you can demand that users select a minimum number of alphabetic, numeric, special, lowercase, or uppercase characters. You can also ensure that the new password selected by a user does not contain, and is not contained by, the password being replaced.

Program Pathing

Program pathing is the ability to demand that a specific file be accessed only through a specific program. Program pathing greatly increases the security of sensitive files. eTrust AC lets you use program pathing to provide additional protection for the files in your system.

B1 Security Level Certification

eTrust AC includes the following B1 “Orange Book” features: security levels, security categories, and security labels.

- Accessors and resources in the database can be assigned a *security level*. The security level is an integer between 1 and 255. An accessor can gain access to a resource only if the accessor has a security level equal to or greater than the security level assigned to the resource.
- Accessors and resources in the database can belong to one or more *security categories*. An accessor can access a resource only if the accessor belongs to all of the security categories assigned to the resource.

- A *security label* is a name that associates a particular security level with a set of zero or more security categories. Assigning a user to a security label gives the user both the security level and any security categories associated with the security label.

Note: For more information about B1 Orange Book features, see the *Implementation Guide*.

Running eTrust AC

eTrust AC can be managed using the Policy Manager interface or a command-line language called selang. These tools let you administer the local workstation and every other Windows workstation on which eTrust AC is installed.

Policy Manager

Policy Manager is the administrative tool for eTrust AC.

selang

The command-line language, selang, performs all the functions of eTrust AC. To use selang commands, open a command prompt window and start selang. You can also use selang in scripts.

For more information about selang and its commands, see the chapter “The selang Command Language” in the *Reference Guide*.

Managing Security for Windows and UNIX

Often large organizations have both Windows and UNIX systems. This complicates the task of maintaining good security. Ideally, you would develop one security policy that can be implemented on both types of systems.

With eTrust AC, you can do all of the following:

- Develop one common security policy for UNIX and Windows
- Implement the policy by using eTrust AC
- Use one Windows workstation to manage the security for Windows and UNIX environments

Making a change and having eTrust AC propagate it to many workstations in different environments can greatly reduce administrative overhead.

Some elements that are particularly important in a common security policy are described in the following sections.

Maintaining One Set of Users

Once eTrust AC is installed at your site, it is possible to maintain one eTrust AC database that contains all the users. This means that user maintenance must be done only once. eTrust AC can propagate the additions, changes, and deletions to all the workstations-both UNIX and Windows-that should receive the updates.

Maintaining One Set of Groups

It is often convenient to group users together who work on specific projects or in specific departments or divisions in the organization. Windows, UNIX, and eTrust AC let you to define groups of users. You can assign authorities to groups just as you would assign authorities to users. Using groups can ease your workload because you assign authorities once for the group rather than repetitively assigning the same authorities to individual users.

Once you are working with eTrust AC, it is possible to create and maintain one set of groups that can be used in both the UNIX and Windows environments.

Maintaining One Set of Access Rules

The Policy Model service lets you develop and maintain one set of access rules for both Windows and UNIX. The PMDB lets you propagate a security database, and any changes made to it, to all its subscribers. Windows and UNIX workstations can be subscribed to the same PMDB.

Communication between the PMDB and its subscribers usually goes in one direction: the PMDB sends changes from its database to its subscribers. A subscriber communicates with the PMDB only when it informs the PMDB that it is online and requests all the changes that were sent by the PMDB while it was down. This design minimizes network traffic, and the integrity of the subscriber is ensured.

Setting Up Administrators

When you installed eTrust AC, you were asked to name one or more eTrust AC administrators. eTrust AC administrators have the authority to modify all or part of the rules database. You should have at least one full-authority administrator. This administrator can modify or create access rules freely and can designate other levels of administrators.

Once you have defined users for your system, you can assign administrative authority to other users by assigning them the ADMIN attribute.

Note: A user with the ADMIN attribute possesses powerful authority. Consequently, the number of ADMIN users should be strictly limited. It is also a good policy to separate the roles of the Windows Administrator and ADMIN, removing the ADMIN attribute from the Administrator after you have set up one or more eTrust AC security administrators.

Because you always need at least one user with authority to manage the database, eTrust AC does not let you delete the last user that has the ADMIN attribute. If you intend to remove the ADMIN attribute from the Administrator, you must first give another user the ADMIN attribute.

If you expect any of the eTrust AC administrators to be administering other hosts from this workstation, be sure that a rule in the database on that host gives them READ and WRITE access from this workstation.

The Best of Both Worlds: Creating Sub Administrators

eTrust AC contains a sub administration feature that allows administrators to grant specific privileges that enable regular users to manage specific classes. These users are then called sub administrators.

For example, you can allow a specific user to manage users and groups only.

You can also specify a higher level of sub administration by granting access not only for specific classes, but for specific *objects* in these classes.

Setting Up Audit Procedures

eTrust AC keeps audit records for events of access denial and access grants according to the audit rules defined in the database. The decision whether to log a certain event is based on the following rules:

- Every accessor and resource has an AUDIT property that can be set to indicate whether access successes, failures, or both, should be logged; in addition, the AUDIT property for accessors can indicate whether login successes, failures, or both should be logged.
- If the resource or the accessor has the AUDIT(ALL) attribute, all events concerning resources protected by eTrust AC are logged, regardless of whether access failed or succeeded.
- If the access to a resource protected by eTrust AC is successful and the user or the resource has AUDIT(SUCCESS), the event is logged.
- If the access to a resource protected by eTrust AC fails and the user or the resource has AUDIT(FAIL), the event is logged.

Only a system auditor, a user to whom the AUDITOR attribute is assigned, can perform auditing tasks such as changing the auditing attribute that is assigned to users and resources.

If a particular resource has been set to warning mode, a violation of access rules for that resource results in an audit record mentioning that the violation was permitted because warning mode is in effect.

The audit records constitute a file called the *audit log* (seos.audit). The location for the audit log is specified in the registry, as is the location for the error log.

The audit log (and also the error log) is specified under the following registry key:

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrustAccessControl\logmgr
```

The audit log is a binary file and cannot be edited or changed. However, you can use Policy Manager to view recorded events, to filter out events by time restrictions or event type, and so forth. (You can also use the seaudit utility to accomplish these same tasks.)

Consider archiving (backing up) old audit logs and error logs to let you scan the events at a later date.

Sending Audit Events to Unicenter TNG

Integration with Unicenter TNG is set up at installation.

You can choose to send audit data to Unicenter TNG, permit launching of eTrust AC from Unicenter TNG, or both. The two options are not interrelated.

Selecting the first option sets registry values under the subkey:

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrustAccessControl\UCTNG
```

The value Integration is set to 1 (yes) and the value EvtManagerServer receives the name of the Unicenter TNG host as a string value.

Audit events that are passed to Unicenter TNG appear in the Console logs in the Unicenter Enterprise Management\Enterprise Managers\Windows NT\Event window.

Audit Event	Display Color	Severity
Success	Blue	S
Denied	Orange	F
Fail	Orange	F
Warning	Blue	W
eTrust AC stopped (audit down)	Blue	I
eTrust AC started (audit start)	Blue	I

The second option permits launching eTrust AC from the Unicenter WorldView menu by pointing to the icon representing the TCP/IP Network in the Managed Objects window and selecting eTrust AC from the right-click menu.

eTrust AC also sends following information about events:

- Product name (eTrust Access Control + version number)
- User name
- Terminal name
- Class name
- Resource name
- Process name
- Event's time
- Full audit message in the format of eTrust AC auditing

The fields User name, Terminal name, Class name, Resource name, and Process name are not always sent, depending on event type.

Using Policy Model Databases

At large sites with many computers and workstations, managing tens or hundreds of eTrust AC databases individually is not practical. When security rules are the same for the majority of computers in an enterprise, administrators need a way to apply security rules once and have those rules propagated as appropriate. In eTrust AC, *Policy Model databases* (PMDBs) provide that facility.

A PMDB contains the same kind of information as a local database (eTrust AC or native operating system), as well as a list of subscribing databases, which can be local databases or other PMDBs that reside on other hosts. A PMDB is essentially a master rules database or template. The rules defined in the PMDB, and any changes made to it, are applied to the subscribing databases.

The PMDB facility provides a simple, hierarchical model for distributing access rules to multiple systems and for creating access rules that are identical for a group of hosts. Configuring PMDBs in a hierarchy lets you support multiple levels of policy, from the top-level template-access rules that apply to all the hosts in an enterprise-to lower level templates that define specific rules applicable to a subgroup of hosts.

Although a PMDB can have multiple subscribers, a PMDB or a local database can be subscribed to a single parent PMDB only for the purpose of receiving propagated rules changes. In addition, each PMDB can have the same parent PMDB or a different one (called a password PMDB), for the purpose of propagating password changes. Password changes, in contrast to rules changes, are propagated in both directions-that is, from the local database on the host on which they originated up to the top of a password Policy Model and back down to all the subscribers in the hierarchy.

Setting Up Encryption

When your network includes more than one server running eTrust AC, communication between the eTrust AC services on the different servers is encrypted. By default, eTrust AC uses a fast and efficient scrambling algorithm for encryption.

eTrust AC also provides AES, DES, and 3DES encryption options that you can choose during the installation.

Standard Encryption

The eTrust AC encryption form is implemented in a dynamic link library (DLL). This DLL implements all data encryption and decryption, permitting data transfer between a client program (selang.exe or SeAM.exe) and the Agent service in a safe form. You can change the default encryption key using the sechkey utility from the command prompt or the ChEncKey.exe utility from Windows.

The eTrust AC installation installs files named defenc.dll (for default encryption), aes128enc.dll (for 128bit AES encryption), aes192enc.dll (for 192bit AES encryption), aes256enc.dll (for 256bit AES encryption), desenc.dll (for DES encryption), and tripledesenc.dll (for 3DES encryption) in the following directory:

eTrustACDir\bin

where *eTrustACDir* is the directory where you installed eTrust AC.

The full path of defenc.dll, aesenc.dll, desenc.dll, or tripledesenc.dll is saved as the registry value Encryption Package in the registry subkey:

HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrustAccessControl

Custom Encryption

If you want to use your own encryption method, you must write a new DLL. The new encryption DLL does not need to be stored in the following directory, but you must replace the registry value Encryption Package with the full path directory name of the new DLL:

eTrustACDir\bin

where *eTrustACDir* is the directory where you installed eTrust AC.

The encryption DLL must contain three exported functions:

1. DWORD Init(PCHAR pKey, DWORD dwLength)

   This function initializes the encryption key.

2. DWORD Scramble(void *param, char *buffWithPlainText, int sizeofbuffWithPlainText, char *buffWithEncryptText, int *sizeofbuffWithEncryptText)

   This function encrypts data received in the second parameter and saves the encrypted buffer in the fourth parameter.

3. DWORD Unscramble(void *param, char *buffWithEncryptText, int sizeofbuffWithEncryptText, char *buffWithPlainText, int *sizeofbuffWithPlainText)

   This function decrypts data received in the second parameter and saves the buffer as plain text in the fourth parameter.

Important! If you change the encryption key or the encryption DLL, you must make the same change in all hosts that communicate with each other. Otherwise, you create a situation in which the encryption is not identical and the hosts cannot communicate successfully.

Chapter 3: Using the Administrator Interface

This section contains the following topics:

The Policy Manager

The Policy Manager is an interface for administering and auditing eTrust AC local databases and PMDBs under Windows and UNIX. The sections in this chapter describe the Policy Manager and illustrate how to use eTrust AC to implement and maintain your security policies. Policy Manager can also administer the native Windows and native UNIX environments. Almost everything that can be done with the Windows User Manager or line commands in UNIX can be done with Policy Manager.

The Policy Manager Interface

All data management begins at the main window of the Policy Manager. This window appears after you successfully log on.

Note: To run Policy Manager, your station must be defined as a management console during installation. For more information, see the *Implementation Guide*.

Menu Bar

The menu bar contains the pull-down menus of commands you can use with Policy Manager. The menu bar structure is dynamic, with appropriate commands appearing for the action you are taking. For example, the Tree menu appears only when the active window contains a tree structure.

Toolbar

The toolbar provides easy access to frequently used commands. Most of the commands are also accessible from the menu bar. Like the menu bar, the toolbar is dynamic, with appropriate commands appearing for the action you are taking. Common tools are described in the following sections. Tools specific to a particular window are described in the section on that functionality.

Program Bar

The program bar lets you choose specific items to protect or to be protected from. To display the panels on the program bar, click the buttons labeled Access Control, Windows NT, and Tools.

Workspace

The workspace displays windows that are opened from the File menu or from the Program Bar. These windows are called application windows.

Output Bar

The output bar displays the command log, which is the file in which eTrust AC writes selang commands. The information shown in the output bar is commands that were created, the host on which they were created, the environment in which they were created, and the date and time they were executed.

Every time you begin a new session of Policy Manager, eTrust AC creates a new command log. Therefore, if you want to save the commands from a session, you should save or print the log.

Note: Each line in the output bar of the Policy Manager window may represent more than one selang command in the Command Log.

Note: For more information about the Policy Manager and how to use it, see the *Policy Manager Online Help*.

Managing Accessors

An *accessor*, sometimes called an account, is an entity that can access system resources. The most common type of accessor is a *user*-typically a person who logs on and for whom access authorities should be assigned and checked. *Groups*, *programs,* and *terminals* are also accessors.

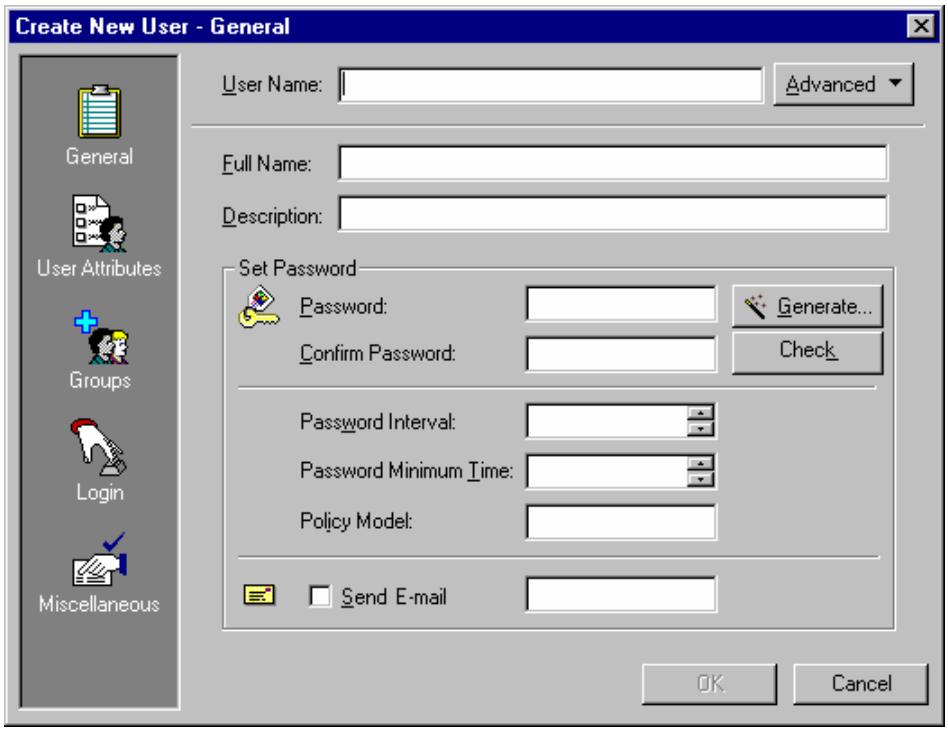
eTrust AC can identify users by account name only or by account name prefixed with a Windows domain name or server name (when the user account is not part of a Windows domain), depending on which you use when you create user records in the eTrust AC database.

You can administer all the users and groups defined in the native Windows operating system and in the eTrust AC database (the eTrust environment). You can do the following:

- Add a user or group to either or both environments (Windows and eTrust).
- Update a user or group in either or both environments.
- Delete a user or group from either or both environments.
- Rename a user (Windows environment only).
- Add a user to or remove a user from a group.
- Add a group to or remove a group from a group.
- View the protected resources of a user or group.

To perform these functions, click  (Users) or  (Groups) in the Access Control panel of the program bar, and then click New, Delete, or Properties on the toolbar.

Here is the dialog for adding a user (click Users and New):



Click the icons on the left to display different panels. For example, the General panel, which is shown, lets you enter user name and description, specify the eTrust AC or Windows environment (Advanced button), and set password information.

Note: eTrust AC also provides wizards for some of the tasks necessary to manage accessors. To access the wizards, click Users or Groups in the Access Control panel, and then choose from the Tools menu or click the Wizards toolbar button.

Important! We strongly recommend that you not use Windows NT backup domain controllers (BDCs) to define users. Most of the functions you can perform in native Windows with the User Manager and User Manager for Domains you can perform in the Access Control and Windows panels of the program bar.

You can import users and groups from your Windows system to the eTrust AC database, during the installation or later using the NT Import Wizard.

Note: For more information and detailed procedures, see the *Policy Manager Online Help*.

Assigning Windows Rights to Accessors

You can assign standard and advanced rights to users and groups in Windows. Most advanced rights are useful only to programmers writing applications for computers running Windows Workstation or Windows Server; advanced rights are not usually granted to a group or end user.

Note: For information about advanced rights, see the Windows Server programming documentation.

Restricting a User Logon

You can restrict user login privileges in several ways:

- Specify an expiration date.
- Suspend an account so that it exists in the eTrust AC database, but the user cannot log on.
- Specify the number of grace logins.
- Specify the maximum number of terminals from which a user can log on.
- Specify the number of days that must pass before an account becomes inactive.
- Limit logon rights to specific days and hours.

By default, the account does not expire or become inactive, the account is not suspended, and a user can log on to any number of terminals without restrictions.

Use the Login panel to restrict login privileges.

Selecting User Activities to Audit

For users defined in the eTrust AC database, you can specify the user activities that eTrust AC should audit.

Note: Only users defined in the database with the AUDITOR attribute can specify audit properties. This option is dimmed for users who are defined in the Native environment only.

The following audit modes specify which user activities are included in the eTrust AC audit log. These options are available from the Miscellaneous panel of the dialogs for creating and editing users.

Success

Successful accesses to resources defined in eTrust AC are logged.

Logon Success

Successful logons are logged.

Logon Failure

Failed logon attempts are logged.

Failure

Failed attempts to access resources defined in the database are logged.

All

All user activity, successful or not, is logged.

None

No user activity is logged.

Entering Personal Information

You can enter personal information about the user from the Miscellaneous panel of the dialogs for creating and editing users. These properties are optional.

Location

An alphanumeric string of up to 128 characters specifying the location of the user, such as Main Office or East Coast Sales.

Country

An alphanumeric string of up to 19 characters indicating the country in which the user is located.

Organization

An alphanumeric string of up to 256 characters indicating the organization to which the user is assigned.

Organization Unit

An alphanumeric string of up to 256 characters indicating the organization unit to which the user is assigned.

Phone

An alphanumeric string of up to 19 characters indicating the user's telephone number.

E-Mail

An alphanumeric string of up to 256 characters indicating the email address of the user.

Setting Account Information

You can set account information for a user from the Miscellaneous panel of the dialogs for creating and editing users. These properties are optional.

Home Directory

The user's home directory. Users log in automatically to their own home directories.

Script

The file name that runs automatically when the user logs in. This login script configures the working environment.

Profile Path

The full path of the file that contains a user's profile. Every time the user logs in to any workstation, the same environment appears on the screen.

Assigning User Privileges

You can assign user privileges from the Miscellaneous panel of the dialogs for creating and editing users. User privileges include items such as:

- Changing the system time
- Loading and unloading device drivers
- Logging in locally
- Restoring files and directories
- Backing up files and directories

Using B1 Security Features

You can add additional security with the B1 “Orange Book” security features. Select security labels, categories, and levels from the B1 Features dialog by clicking the B1 Features button on the Miscellaneous panel of the dialogs for creating and editing users.

- You can assign users a security level between 1 and 255. The user can gain access to a resource only if the user has a security level equal to or greater than the security level assigned to the resource.
- Users can belong to one or more security categories. A user can access a resource only if the user belongs to all of the security categories assigned to the resource.
- Security labels associate a particular security level with a set of security categories. Assigning a user to a security label gives the user both the security level and any security categories associated with the security label.

Assigning a Session Group

You can assign an eTrust SSO session group to a user using the Session Group button on the Miscellaneous panel for creating and editing users.

Adding a User to a Group

You can add users to a group to make managing them much easier. Use the Groups panel of the dialogs for creating and editing users.

Adding Nested Groups

You can add or modify nested groups from the Miscellaneous pane of the dialogs for creating and editing groups. (Click Groups in the program bar and then New or Properties on the toolbar.)

The Nested Groups dialog lets you add and delete super groups (parents) and member groups (children) from existing groups. Properties of a super group are passed down to its member groups.

Setting Active Directory Properties

When you are connected to a Windows 2000 computer with Active Directory, you can use the Directory Services panel of the User or Group Properties dialog to set Active Directory user or group properties. These properties are not supported in Windows NT, Windows 2000 without Active Directory, or the eTrust AC native environment database.

The icon to activate the panel does not appear unless you are connected to a Windows 2000 machine with Active Directory.

Note: Active Directory lets you organize users into different folders. Policy Manager displays all Active Directory users in a single Users panel.

Synchronizing Data with the Native Operating System

When using selang commands, you can change data about an accessor in the database without changing data in the native operating system. Likewise, when using the User Manager in Windows, you can change data about an accessor in Windows without changing data in eTrust AC. When you change data in either of these ways, the accessor is defined differently in each database.

eTrust AC monitors definitions in eTrust AC and the native operating system, and provides a Synchronization panel when the definitions in Windows and eTrust AC do not match. When the definitions match, the Synchronization icon is not visible.

Managing eTrust AC Resources

A *resource* is an entity that users and groups can access. The most common type of resource is a file. You access a file when you read information from it or write information to it.

Resources are grouped by *class*, which is a name for the type of resource. For example, the TERMINAL class contains all objects that are terminals, such as tty1, tty2, and so on; the SHARE class contains all objects that are shared; the FILE class contains definitions for files and directories.

Note: For more information about the eTrust AC classes, see the *Reference Guide*.

The properties of a protected resource are stored in the resource's *record*. A record is a collection of data consisting of the name and properties of a resource. Every record in a particular class contains values for the same set of properties-the properties appropriate to the type of object that the class describes.

Properties indicate who defined the resource, the date when the resource was defined, and more. In general, the most important information contained in a resource record is the list of accessors authorized to access the resource. This list is called the access control list (ACL). Many resources contain another list of accessors, for which access is denied. This list is called the negative access control list (NACL).

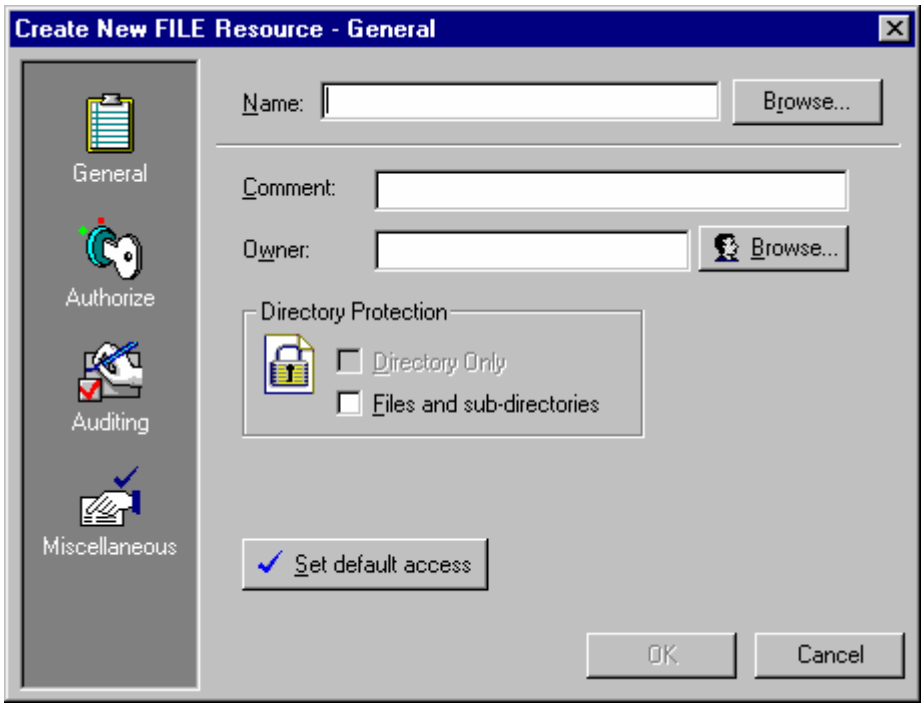
Note: You can view the ACLs or NACLs for a specific user or group by choosing Protected Resources from the menu displayed when you right-click a user or group name.

You can administer all the resources in the eTrust AC database by doing the following:

- Add a resource to any class in the eTrust AC database
- Update a resource in any class in the eTrust AC database
- Delete a resource in any class from the eTrust AC database
- Define terminals and terminal groups from which users can log on
- Define holidays when users need extra privileges to log in
- Define task delegation and task groups

To perform these functions, click  (Resources) in the Access Control panel of the program bar, select a resource in the workspace, and then click New, Delete, or Properties on the toolbar.

Here is the dialog for creating a resource in the FILE class (click Resources and New):



Click the icons on the left to display different panels. For example, the General panel, which is shown, lets you enter resource name and description, specify the owner, and more.

Using the Calendar to Manage eTrust AC Resources

eTrust AC supports user, group, and resource access enforcement according to the Unicenter TNG calendar. The calendar contains time intervals of 15 minutes that you can set to ON or OFF. A calendar time interval set to OFF prevents access to resource; a calendar time interval set to ON allows access to the resource. eTrust AC retrieves Unicenter TNG active calendars at specified time intervals.

You can add, edit, or remove a calendar resource using the Resources view. Select Login Protection in the resources tree. Click the Calendar tree entry, and right-click to select an option.

Managing Windows Resources

You can administer the resources in the native Windows database using the dialogs for creating and editing resources. You can:

- Add a resource to the REGISTRY and SHARE classes in the Windows database.
- Update a resource in any class in the Windows database, including the Active Directory database.
- Delete a resource in any class from the Windows database.

Note: For more information about Windows resources, see the Windows environment classes and properties in the *Reference Guide*.

Managing Windows Domains

Using Policy Manager, you can:

- Display information about a Windows domain.
- Add new computers to a Windows domain.
- Delete computers from a Windows domain.
- Create and delete trusted relationships between Windows domains.

Select NT Specific in the resources tree. Click the Domain tree entry, and right-click to select an option.

eTrust AC checks the validity of these operations if an eTrust AC client, such as Policy Manager or selang, performs them. When it checks the validity of an operation, eTrust AC uses the authorization rules that exist in the eTrust AC database in the domain controller.

Each record in the eTrust AC class DOMAIN defines a Windows domain. Three types of possible access for records in the DOMAIN class are:

READ

Lets the user display the properties of the domain

CHMOD (Change Trust)

Lets the user create or delete trust relationships between domains

EXEC (Execute)

Lets users add members to or delete members from a domain

Protecting Processes

Objects in the PROCESS class define process applications that need eTrust AC protection.

To protect processes with eTrust AC, complete the following steps:

1. Start the eTrust AC Policy Manager.
2. From the Program bar, select Resources.
3. Expand the eTrust AC Resource tree to display System Resources and select Process.
4. To add new processes to protect, right click the name column, and select New or click Insert on your keyboard.
5. Start Windows Task Manager (taskmgr.exe). The Task Manager must be running to perform the steps that follow.
6. Click Browse next to the Name field. Select a Process to protect (taskmgr.exe). Click OK.
7. Click Browse next to the Owner field.
8. Select nobody. Click OK.
9. Click the Set Default Access button.

   The Set Default Access dialog appears.
10. Ensure that None is selected. Click OK.
11. To test the rule, open the Task Manager (taskmgr.exe), select the Processes Tab, select taskmgr.exe, and click End Process.
12. A Task Manager Warning dialog box will appear. Click Yes.

    If the rule has executed successfully, a dialog box titled Unable to Terminate Process will appear.
13. To add authorized users to end selected processes, expand the eTrust AC Resource tree to display System Resources and select Process.
14. Right click a Process and select Properties.
15. Click Authorize.
16. Click Insert.
17. Click Browse next to the Name field.
18. Select the Users or Groups tab and click OK.
19. Check Read Permission and click OK.
20. To test the new rule, log in as the user authorized to end the process.

21. Open the Task Manager (taskmgr.exe), select the Processes tab, select taskmgr.exe, and click End Process.

    The process should end.

Note: Windows services are good candidates for eTrust AC process protection because most of these services run in the background instead of GUI or interactive applications.

Protecting Resources with SPECIALPGM

Objects in the SPECIALPGM class define an application that needs special eTrust AC authorization protection. This class is especially useful for protecting programs, such as system services, that must typically be run under the System account. To protect such programs, define them as records in the SPECIALPGM class and associate a logical user name (defined as a USER record in the eTrust AC database) with the Windows user name required to run the program, authorizing only that logical user to run the programs.

In Windows, you can use the Special Program Wizard to help set up this protection. To run this wizard from the GUI, click the Resources button in the program bar. Then select Special Program Wizard from the Tools menu.

Managing Policy Models

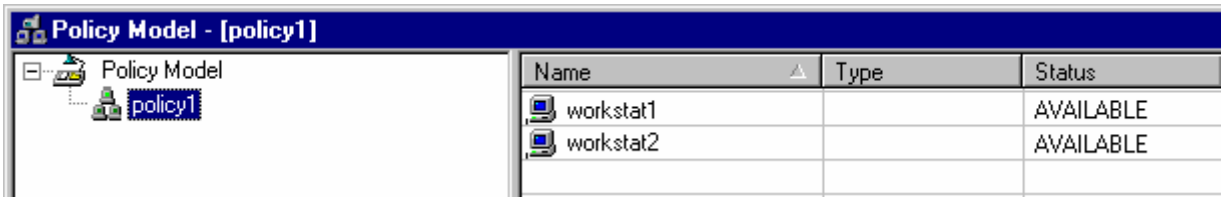
You can use Policy Manager to manage several PMDB functions. These include specifying a PMDB, managing subscribers, managing the error log, starting and stopping the Policy Model daemon (in UNIX), reactivating an unavailable subscriber, and displaying properties.

Specifying the PMDB

eTrust AC supports multiple Policy Models on a single host. You can specify the PMDB using Policy Manager or selang.

Displaying the Policy Model Window

The Policy Model window, activated from the Tools panel of the program bar, lists all the PMDBs defined on the station to which you are connected, including subscribers where applicable.



The Policy Model window contains the following columns:

Name

Lists the subscribers of the selected PMDB.

Type

Displays the type of subscriber: eTrust database, PMDB, or MF (mainframe).

Status

Indicates whether the subscriber is Available or Unavailable. A subscriber is Available when no commands are waiting to be executed. A subscriber is Unavailable if its parent PMDB has sent one or more commands that have not yet been executed. Commands are saved in the file updates.dat, whose default location is *eTrustACDir*\data\pmdb (where *eTrustACDir* is the directory where you installed eTrust AC).

Next Command

Displays the command that is waiting to be executed.

If the subscriber's status is Available, this column is empty.

Errors

Displays the number of errors for the selected subscriber. An error is a command that failed; that is, it did not update the subscriber. Connection failures are not included.

Executed Commands

Displays the percentage of commands that have been executed. If the subscriber's status is Available, this column displays the value 100%.

Managing the Policy Model Hierarchy

Subscribers to a PMDB can be:

- Another PMDB on the same or a remote host
- An eTrust database on the same or a remote host
- A mainframe database

Using Policy Manager, you can:

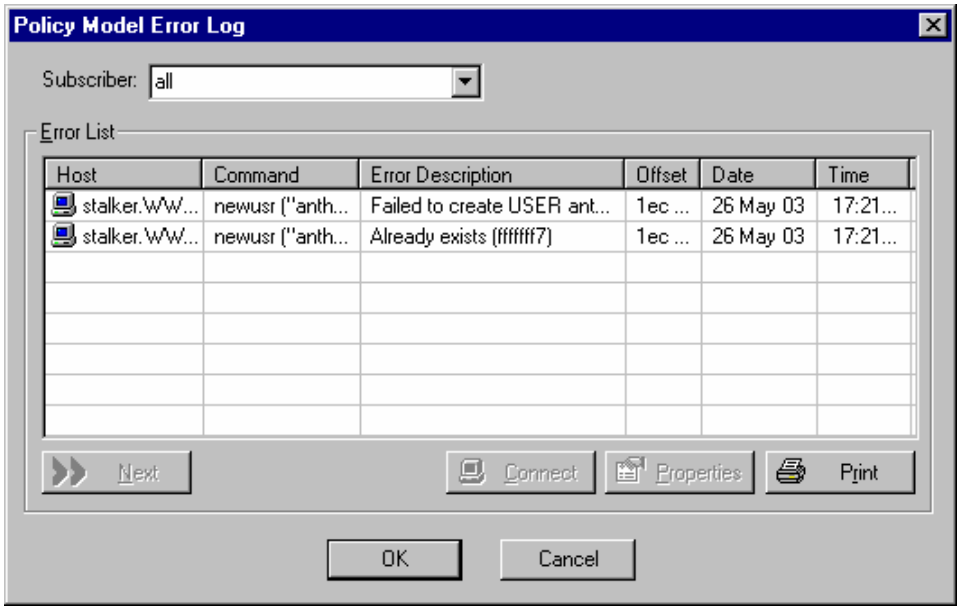
- Add subscribers to a PMDB
- Remove subscribers from a PMDB
- Display the commands that were sent to subscribers but failed to update them-the errors that appear in the error log
- Erase the contents of the error log

When adding a subscriber, ensure that the parent PMDB and all the stations you want to subscribe to it are part of the same network and can communicate with each other by name. This lets eTrust AC update the parent_pmd key in the registry of the subscriber.

Working with the Error Log

The Policy Model error log contains a list of transactions that the subscriber stations refused to apply.

Using Policy Manager, you can display the errors of a PMDB and all its subscribers, or you can display errors for only one subscriber. You can also clear the contents of the error log.



The Policy Model Error Log dialog contains the following columns:

Host

The full name of the PMDB on which the command failed.

Command

The full eTrust AC command that failed.

Error Description

The reason why the command failed.

Offset

The location of the command in the updates.dat file.

Date

The date on which the command failed.

Time

The time the command failed.

Note: If you click the Next button, eTrust AC brings the next set of records. The query_size registry key defines the number of records in a set. (The default value is 100.) The records in the next set are added to the display. This means that if you pressed Next once (and the value of the key is still 100), then 200 records display.

Displaying Properties

Display the properties of a PMDB or a subscriber by selecting Properties from the View menu or the right-click menu.

The descriptions of the properties displayed for the parent PMDB are as follows.

Policy Model Name

The name of the PMDB.

Parent Policy Model

Indicates whether the PMDB is a parent.

Password File

For UNIX only, the name of the file that contains information about the locally defined users such as their full names, IDs, the ID of the groups to which the users belong, their home directories, and encrypted passwords.

Group File

For UNIX only, the name of the file that contains information about the locally defined groups such as the group IDs and the list of users in the groups.

eTrust AC displays the Policy Model window (see page 53) to show the properties of subscribers.

Managing UNIX with eTrust AC for Windows

You can use Policy Manager to manage UNIX machines on which eTrust AC is installed. You can manage users, groups, resources, and PMDB hierarchies defined in the UNIX and eTrust AC environments.

Administrator Resources

ADMIN Class

You can add objects to the ADMIN class by using the Access by Class feature of the Policy Manager.

- The ADMIN class contains the definitions that allow non-ADMIN users to administer specific classes.
- An ADMIN record represents each eTrust AC class that will be administered by delegated users.
- The record contains a list of accessors with the access authorities for each record.
- The key of the ADMIN class record is the name of the class that will be protected.

Example

To define non-ADMIN users to perform ADMIN class functions, us the following example as a guide:

Note: The following example requires two workstations to complete. We'll call the local terminal **computer1** and the remote terminal **computer2**. Computer names may vary in your environment.

1. On the local terminal (computer1) create a user called sub_admin.
   a. Start eTrust AC.
   b. On the program menu click User and create a new user.
   c. In the User Name field enter sub_admin.
   d. In the Full Name field enter sub_admin.
   e. In the Description field enter sub_admin.
   f. In the Password field enter a password that you will remember.
   g. Click the advanced drop down menu. Make sure to check both:
      – Create in Native OS environment
      – Create in eTrust AC environment
   h. Click OK

2. On the remote terminal (computer2) create a user called sub_admin.
   a. Start eTrust AC.
   b. On the Program menu click User and create a New User.

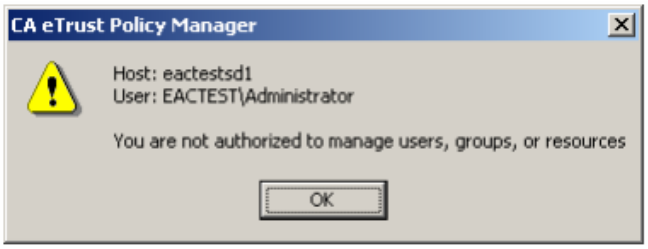
      Note: The remote access user name must be entered in the following format:

      `computer_name\user_name.`

   c. In the User Name field, enter computer1\sub_admin.
   d. In the Full Name field, enter computer1\sub_admin.
   e. In the Description field, enter computer1\sub_admin.
   f. Click the Advanced drop down menu. Choose only create in eTrust AC Environment. Click OK.
3. On the remote terminal (computer2) allow the local terminal (computer1) access.
   a. On the Remote Terminal (computer2), click the eTrust AC Program menu and click Resources.
   b. Expand the login protection tree, right click an open field in the Name column, and select New.
   c. Enter your Local Terminal (computer1) information.
   d. In the Name Field enter the name of the Local Terminal (Computer1).
   e. In the Comment Field enter Terminal (Computer1).
   f. In the Owner Field enter *nobody*.
   g. Click Set Default Access and make sure that *None* is selected.
   h. Click OK.
   i. Click Authorize.
   j. Click Insert.
   k. Click the Browse button next to the Name field.
   l. Select the computer1\sub_admin user that you created earlier and click OK to return to the Add/Edit eTrust Accessor dialog.
   m. An Add/Edit eTrust Accessor dialog box appears. Click OK.
   n. Select the User you just added, check the read Permission, and click OK.

4. On the remote terminal (computer2), expand the Administration tree, select Access by Class, and select the USER Class.
   a. Right click the USER Class, select Properties, and click Authorize.
   b. Click Insert.
   c. Click Browse next to the Name field.
   d. Select the computer1\sub_admin user you created earlier and click OK to return to the Add/Edit eTrust Accessor dialog.
   e. The Add/Edit eTrust Accessor dialog box appears. Click OK.
   f. Select the User you just added, check the read Permission, and click OK.
5. Make sure that both computers have eTrust AC running on the same network.
6. On the Local Machine (computer1), log into windows as the sub_admin user.
7. On the Local Machine (computer1), startup eTrust AC. Click the connect icon located on the upper left.
8. Enter the name of the remote terminal host (computer2) and click OK.
9. A Connection Information dialog appears. Click OK.

   After you click OK, the Policy Manager will open. You will be connected to the Remote Terminal (computer2).
10. In the Policy Manager, click  (Users).
11. The following error message should appear.



12. Access by Class is also used for sub-administration.

    To allow sub-administrators to manage administrator-specified classes and resources, complete the following steps on the remote terminal (computer2) only:
    a. Select the Tools menu, Options, and the Startup tab.
    b. Check Enable Users & Groups Sub Administration and click OK.

13. Click USERS again.

    You will see the User List and User Properties. If you try to change any User Property, an error message will appear that states, “Operation not allowed.” This occurs because the user has Read access only in the ADMIN/USER class.

    The “Operation not allowed” error message will also appear if you click Groups because the user is not an authorized user for the ADMIN class or GROUP class.

14. To browse for Groups and Files in the Group Properties dialog box, you must add Read privileges to the Object Group in the Set Admin Properties dialog box.

    a. From the Remote Terminal, expand the Administration Resource tree, select Access by Class, right click the Group Class, and select Properties.

    b. Click Authorize, click Insert, select the terminal that you created, and set the User Permissions.

    c. From the Local Terminal connect to the Remote Terminal and click Group.

    d. You should see the Group Class.

Container Class

Each record in the Container Class defines a group of objects, files, and users from other resource classes. This simplifies the definition of access rules when a rule applies to several different classes of objects. Members of a Container class record can be objects, files, and users from any of the eTrust AC classes.

Example

In this example you create an object named MyContainer in the CONTAINER class. Gather two members for this object, one from FILE class (c:\winnt\notepad.exe) and the other from the TERMINAL class (MyComp.ca.com). You then authorize or allow Read and Execute privileges for John to MyContainer.

1. Expand the System Resource tree, right click File, click New, and enter the following information:

    a. In the Name field, enter c:\winnt\notepad.exe.

    b. In the Comment field, enter Container Test Rule.

    c. In the Owner field, specify Nobody

    d. Set Default Access to None.

2. Expand the Administration Resource tree, right click Container, select New, and enter the following information:

   a. In the name field, enter MyContainer.

   b. In the Comment field, enter MyContainer test.

   c. In the Owner field, specify nobody.

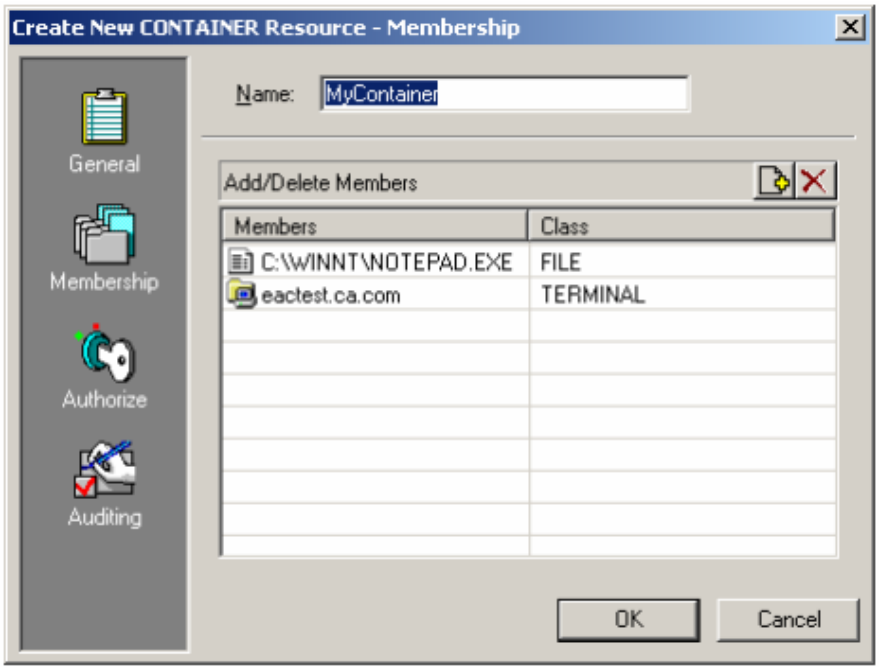
3. Click Membership, right click an open field in the MEMBERS column, and click Add.

4. Select the File class, highlight the File rule that you created (c:\winnt\notepad.exe), and click OK.

5. Right click an open field in the Members column and click Add.

6. Select the Terminal Class, select the Terminal Name that you would like to add, and click OK.

   Your Container Resource dialog should look like this:

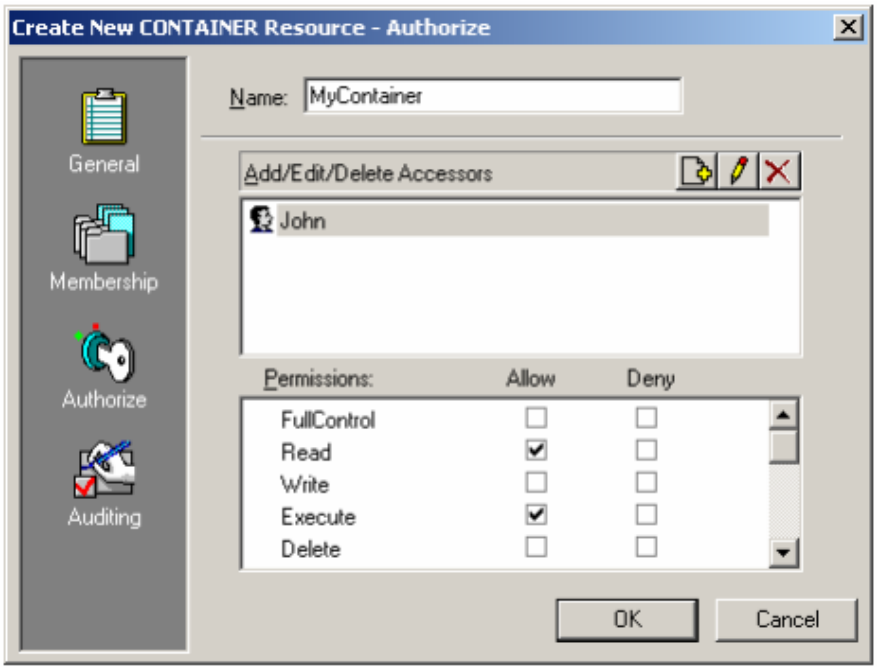


7. Click Authorize, click Insert, click Browse next to the Name field, and select User. Then click OK.

   The Add/Edit eTrust Accessor dialog appears.

8. Click OK.

9. Authorize the user that you select with Read and Execute Permissions.



10. Log in as the user specified earlier. You should login successfully since you allowed Read access to this user in MyContainer.
11. Run the c:\winnt\notepad.exe and allow it to execute.
12. Try to delete c:\winnt\notepad.exe. This action should be prohibited.
13. Try to open a session for an eTrust AC client, the Policy Manager, or selang. The action should be prohibited because John was not allowed Write access in MyContainer.
14. Try to change your User Permissions and deny all items.

    The specified user should be denied login. The execution of c:\winnt\notepad.exe should also be denied.

Creating Sub Administrators

To set up sub administrators to manage users and groups from the Policy Manager, complete the following steps:

1. Launch Policy Manager.

   Note: If eTrust AC server is installed on this machine, shut down eTrust AC services after you log in to Policy Manager.

2. From the Policy Manager toolbar, select Tools, Options.

   The Options dialog displays.

3. Select the Startup tab, then check Enable Users and Groups Sub Administration.

4. Click OK.

To enable sub administrators to access Policy Manager from a specific terminal, complete the following steps:

1. Select the Resources icon in the eTrust AC program bar to display the Resources window.

2. Expand the Login Protection folder.

3. Select Terminal to display the list of available terminals.

4. Double-click the terminal you want. The View or Set Terminal Properties - General dialog displays.

5. Select the Authorize icon to display the View or Set Terminal Properties - Authorize dialog.

6. Select the sub administrator you want to authorize and check Read and Write permissions.

7. Click OK.

To define a sub administrator with privileges to manage users, do the following:

1. Select the Resources icon in the eTrust AC program bar to display the Resources window.

2. Expand the Administration folder.

3. Select Access by Class to display the list of available classes.

4. Double-click the USER class and choose Properties. The View or Set ADMIN Properties - General dialog displays.

   Note: To enable the sub administrator to administer other classes, replace the USER class with the class you want (GROUP, USER_DIR, and so forth).

5. Select the Authorize icon to display the View or Set ADMIN Properties - Authorize dialog.

6. Click Add to display the Add eTrust AC Accessor dialog.
7. Enter the name of the sub administrator in the Name field or click Browse to locate.
8. Check the permissions you want to give the sub administrator access to.
9. Click OK to return to the View or Set ADMIN Properties - Authorize dialog.
10. Click OK to finish.

Chapter 4: Managing User Passwords

This section contains the following topics:

Password Management Utilities

You can use the following to manage user passwords:

Password Manager

You can use the Password Manager to set and replace user passwords.

The Password Manager is an independent utility for password management that can be installed on machines not running Policy Manager. This facilitates assigning password administration tasks to non-administrators.

Policy Manager

Each time you create or update users defined in Windows, you can set or replace user passwords.

You also use Policy Manager (see page 39) to set password policies.

selang commands

The selang commands newusr, editusr, and chusr set a user's password.

Note: For more information about these commands, see *Reference Guide*.

Windows utilities

You can use the Windows User Manager or Windows commands in a command prompt window to manage user passwords.

Note: For more information, see the relevant Windows documentation.

When you use the Password Manager, Policy Manager, or selang to set or change a user's password, eTrust AC does not check its password rules before adding the password to the database. eTrust AC accepts any password that you enter from these facilities. As a result, the new password may not be valid according to the password rules in eTrust AC. When you use the Windows User Manager or other non-eTrust AC software to change a password, eTrust AC checks the validity of the new password.

Managing Password and Lockout Policies

Passwords are the most popular device for authentication, but password protection methods have well-known problems: trivial passwords are easy to guess; passwords that last for years and cyclic passwords are eventually broken; and passwords sent in clear text over a network can be trapped by listeners.

Windows has a set of password rules and policies that force users to use passwords that avoid most of these common pitfalls. eTrust AC has additional rules that ensure that users select even more secure passwords.

You can specify the following rules in eTrust AC:

- A new password cannot match previous passwords. The number of previous passwords that eTrust AC stores is specified in the password policy.
- A new password cannot contain the user name.
- A new password cannot contain the password that it is replacing.
- A new password cannot match the password that it is replacing. eTrust AC disregards letter case.
- A new password must have at least the minimum number of alphanumeric characters, special characters, digits, lowercase characters, and uppercase characters specified in the password policy.
- A new password must not have more repetitive characters than is specified in the password policy.
- A new password cannot be one of the restricted words in the dictionary included in eTrust AC. The dictionary is specified in the Dictionary value in the registry subkey:

  `HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrustAccessControl\passwd`

Each password must have a maximum lifetime; that is, it must expire, forcing the user to choose a new password after a certain interval.

- Each password must have a minimum lifetime. By specifying a minimum lifetime, you can prevent users from quickly and repeatedly changing passwords. With frequently changed passwords, they could overflow the password history stack and then re-use a previous password.

Using the Password Manager

If the Password Manager was installed on your station, you can use it to set new passwords or replace existing passwords.

You can do the following when you set or replace passwords:

- Require users to change their passwords the next time they log on.
- Reset a locked user account so that the user is no longer locked out.
- Change the target host so that you can set passwords for users on a remote host, on a PMDB on the local host, or on a PMDB on a remote host. By default, eTrust AC assumes the users are on the local host.
- Configure eTrust AC to generate user passwords that meet the password policy of your organization.
- Configure eTrust AC to send email to users whose passwords you change to notify them of their new password.

Generating Passwords

You can generate passwords for existing users in the Password Manager. The passwords generated by eTrust AC always meet the criteria you have established for your site. To generate passwords you must select Enable Password Generation as a system option. By default, the password generation option is selected.

Changing the Target Host

You can change the target host so that you can set passwords for users on a PMDB on the local host, on a remote host, or on a PMDB on a remote host.

Setting Up User Password Changes

In Policy Manager, you can set up a facility to remind users to change their passwords. When they log in to the Policy Manager, a dialog appears reminding them to change their password. If they click Yes, the Change Password dialog appears.

To set up this feature, complete the following steps:

1. Click the Resources icon on the Access Control program bar.
2. Select Activate eTrust Classes from the Tools menu. Check the Password box and also the Change Own Password box.
3. Click OK.

Resolving Error Messages

If you are setting passwords for users on Windows NT systems, the following message may appear:

```
The password is shorter than required.
```

This error means that the password does not meet the policy requirements. This is caused by any of the following:

- The password is shorter or longer than the required length.
- The password has been used recently and exists in the Windows NT Change History field.
- The password does not have enough unique characters.
- The password does not meet other password policy requirements (such as those set with eTrust AC password policies).

To avoid this error, make sure you set a password which meets all applicable requirements.

Chapter 5: Protecting Accounts

This section contains the following topics:

Protecting User Impersonation Requests

After an account logs in, you must monitor it to ensure that it performs only authorized functions on system resources. The operating systems provide some degree of protection for files based on the accessor's user SID. To bypass that protection, a user must first impersonate himself to another user SID. The operating system's protection against unauthorized impersonation is that the impersonation action asks the requesting user to specify the target user's password.

This scheme has many faults. A user who wants to impersonate himself to a user SID must memorize the target user's password, write it down, or ask the target user to use a trivial password. This violates several password policies. In addition, there is no effective accountability; you can never tell which user changed identification to a specific user. Moreover, once the password of the super-user is known to a user, any security is bypassed, and that user has unlimited access to the system.

eTrust AC uses a more advanced method for protecting impersonation: a user can change the user SID to another user's SID only if a specific rule allows the change.

For example, suppose that user X runs a program that performs some tasks as user Y. User X has user Y's password but does not have permission to substitute to user Y, so the program request is denied.

This way, Administrator's password is not enough for an intruder; there must also be a rule in the database that allows the intruder to become Administrator.

Each user SID and group SID can have an access rule in the database. eTrust AC has assigned the SURROGATE class for this type of protection. If in the initial stage you wish to grant access to any impersonation request, use the following command:

```
eTrust> editres SURROGATE _default defaccess(READ)
```

This command tells eTrust AC to allow access if a user makes a request to impersonate himself to another user, and a record in the database does not explicitly protect the user substitution.

To protect against attempts to substitute the SID to that of the super-user, use the following command:

```
eTrust> newres SURROGATE USER.Administrator defaccess(NONE)
```

This command tells eTrust AC that the Administrator user name is protected and that users not explicitly permitted to use it cannot impersonate to Administrator. To permit the security administrators to use Administrator, you must explicitly specify it by using the following command:

```
eTrust> authorize SURROGATE USER. Administrator gid("Security Admins")
```

Notes:

- If a SURROGATE record of a user does not specifically permit a certain user to do the substitution, the user gets the default access of that record. In the previous example, the default is NONE, which means that users without permission cannot impersonate to Administrator.
- A record called USER._default represents all users who do not have their own records. Similarly, record called GROUP._default represents all groups that do not have their own records. If no SURROGATE record for a certain accessor exists, a request for substitution to that accessor ends with the default specified in the SURROGATE USER._default record, the SURROGATE GROUP._default record, or the _default record (for both users and groups).
- The default value for the _default record is READ; undefined SURROGATE records imply permission to impersonate into those users. This default meets the general rule of thumb during implementation that says “whatever is not defined in eTrust AC is not protected by eTrust AC.” You can modify this rule after the implementation stage to the opposite rule: “whatever is not permitted in eTrust AC is automatically forbidden by eTrust AC.”
- Many Windows utilities and services (for example, Run As) identify as user "NT AUTHORITY\SYSTEM" and not as the original user running them. To let users who use these utilities and services as impersonate another user, you must create this SYSTEM user in the eTrust AC database and authorize it to impersonate the target user. For example:

  ```
  auth SURROGATE USER.Administrator uid("NT AUTHORITY\SYSTEM") acc(R)
  ```

Setting Up the Surrogate DO Facility

Operators, production personnel, and end users often need to perform tasks that only the super-user can perform.

The traditional solution is to supply all these users with the super-user's password, which compromises the security of the site. The secure alternative - keeping the password secret - results in the system administrator being overloaded with legitimate requests from users to perform routine tasks.

The Surrogate DO (sesudo) utility solves this dilemma. It allows users to perform actions that are defined in the SUDO class, where each record contains a script, specifies which users and groups can run the script, and lends them the necessary permissions for the purpose.

For example, to define a SUDO resource that starts the "Print Spooler" service as if the user were System, enter the following command:

```
eTrust> newres SUDO StartSpooler data("net start spooler")
```

This newres command defines StartSpooler as a protected action that some users may receive System authority to perform.

Important! In the data property, use a full absolute path name. A relative path name could accidentally execute a Trojan horse program planted in an unprotected directory.

In addition, users can be authorized to perform the StartSpooler action by using the authorize command. For example, to allow the user *operator1* to start the "Print Spooler" service, enter the following command:

```
eTrust> authorize SUDO StartSpooler uid(operator1)
```

You can also explicitly prevent a user from performing the protected action by using the authorize command. For example, to prevent the user *operator2* from starting the "Print Spooler" service, enter the command:

```
eTrust> authorize SUDO StartSpooler uid(operator2) access(None)
```

Executing the sesudo utility performs the protected action. For example, the user *operator1* would start the "Print Spooler" service using the following command:

```
cmd> sesudo -do StartSpooler
```

The sesudo utility first checks whether the user is authorized to perform the SUDO action and then, provided the user is authorized to the resource, executes the command script defined in the resource. In the case of our example, sesudo checks whether operator1 is authorized to perform the StartSpooler action and then invokes the command "net start spooler" with System credentials.

Note: For more information about the sesudo utility, see the *Utilities Guide*. For more information about formatting the SUDO record's data property, see the chres, editres, and newres commands in the *Reference Guide*.

Checking User Inactivity

The inactivity feature protects the system from unauthorized access through accounts whose owners are away or no longer employed by the organization. An inactive day is a day in which the user does not log in. You can specify the number of inactive days that must pass before the user account is suspended and cannot log in. Once an account is suspended, you must manually reactivate it.

Note: Password changes count as activities, in terms of inactivity checks. If a user's password changes, that user cannot become suspended due to inactivity.

You can set the number of inactive days with the inactive property of a USER class record or a GROUP class record. The latter affects only users that have that group as a profile group. You can also set inactivity for all users system-wide with the INACT property of the SEOS class.

Both selang and the Security Administrator provide the means for setting inactivity. In selang, use the following command to specify inactivity globally:

```
setoptions inactive ( numdays)
```

To set the number of days for a group (which overrides the system-wide inactive setting for that group), use the following command:

```
{chgrp | editgrp | newgrp} groupName inactive ( numdays)
```

To set the number of days for a user (which overrides group and system-wide settings for that user), use the following command:

```
{chusr | editusr | newusr} userName inactive ( numdays)
```

To reactivate a suspended user account, use the following command:

```
{chusr | editusr} userName resume
```

To reactivate a suspended profile group, use the following command:

```
{chgrp | editgrp} userName resume
```

To disable inactive login checking at the system-wide level, use the following command:

```
setoptions inactive-
```

To disable inactive login checking for a group, use the following command:

```
{chgrp | editgrp} groupName inactive-
```

To disable inactive login checking for a user, use the following command:

```
{chusr | editusr} userName inactive-
```

Chapter 6: Managing Policies Centrally

This section contains the following topics:

The Policy Model Database

Managing tens or hundreds of databases individually is not practical. eTrust AC supplies the Policy Model service, a component that lets you manage many databases from one central database. Using the Policy Model (PMD) service is optional, but it greatly simplifies administration at large sites.

Note: In Windows Task Manager, the Policy Model service appears as sepmdd.exe.

The Policy Model service, uses a Policy Model database (PMDB). Like other eTrust AC databases, the PMDB contains users, groups, protected resources, and rules governing access to the resources. In addition, the PMDB contains a list of *subscriber* databases. Each subscriber is an eTrust AC database that resides on a separate computer, or another PMDB that resides on the same or another computer. A PMDB that updates a subscriber is the subscriber's *parent*.

The PMDB is a useful tool for managing many databases that have similar authority restrictions and access rules.

Note: For information about administrating PMDBs using the sepmd utility and managing PMDBs remotely, see the *Reference Guide*.

PMDB Location on Disk

All PMDBs on a computer reside in a common directory. The _pmd_directory_ value in the following subkey of the Windows registry specifies the name of the directory:

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrustAccessControl\Pmd
```

The default value of _pmd_directory_ in the NTFS root directory is: *eTrustACDir*\data, where *eTrustACDir* is the directory where you installed eTrust AC (by default C:\Program Files\CA\eTrustAccessControl).

Each PMDB occupies a subdirectory in the common directory. The files in the subdirectory contain all the data required to define the Policy Model. Policy Model configuration settings are stored in a Pmd sub-key of the eTrust AC registry settings. The name of the sub-key is the name of the Policy Model.

Managing Local PMDBs

eTrust AC offers a utility for administering PMDBs:

sepmd

A PMDB administration utility that lets you:

- Administer subscribers
- Truncate the update file
- Administer Dual Control
- Manage the Policy Model log file
- Perform other administrative tasks

Note: For a comprehensive discussion of sepmd, see the *Reference Guide*.

Managing Remote PMDBs

eTrust AC also offers you a range of selang commands that you can use in the pmd environment. These commands let you manage PMDBs remotely:

createpmd

Creates a PMDB.

deletepmd

Deletes a PMDB.

findpmd

Displays the names of all PMDBs on the computer.

listpmd

Lists the following information about a PMDB:

- Subscribers and their status
- PMDB description and its status
- Commands in the update file and their offsets
- Contents of the error log

pmd

A PMDB administration command that lets you:

- Remove a subscriber from the list of unavailable subscribers
- Clear the Policy Model error log
- Start and stop the Policy Model service
- Truncate the update file
- Reload the registry settings

subs

A PMDB subscription command that lets you:

- Add a subscriber to a parent PMDB
- Assign a parent PMDB to a database (eTrust AC or another PMDB)

subspmd

Assigns a parent PMDB to the local database.

unsubs

Removes a subscriber from the PMDB.

Note: For a comprehensive discussion of selang commands you can use in the pmd environment, see the *Reference Guide*.

Architecture Dependency

When deploying eTrust AC, you should consider the hierarchy of your environment. At many sites, the network includes a variety of architectures. Some policy rules, such as the list of trusted programs, are architecture-dependent. On the other hand, most rules are independent of the system's architecture.

You can cover both kinds of rules by using a hierarchy. You can define a global database for architecture-independent rules, and give it subscriber PMDBs that define architecture-dependent rules.

Note: The root PMBD and all of its subscribers can reside on the same computer or on separate computers, depending on the physical needs of your environment.

Example: A Two-tiered Deployment Hierarchy

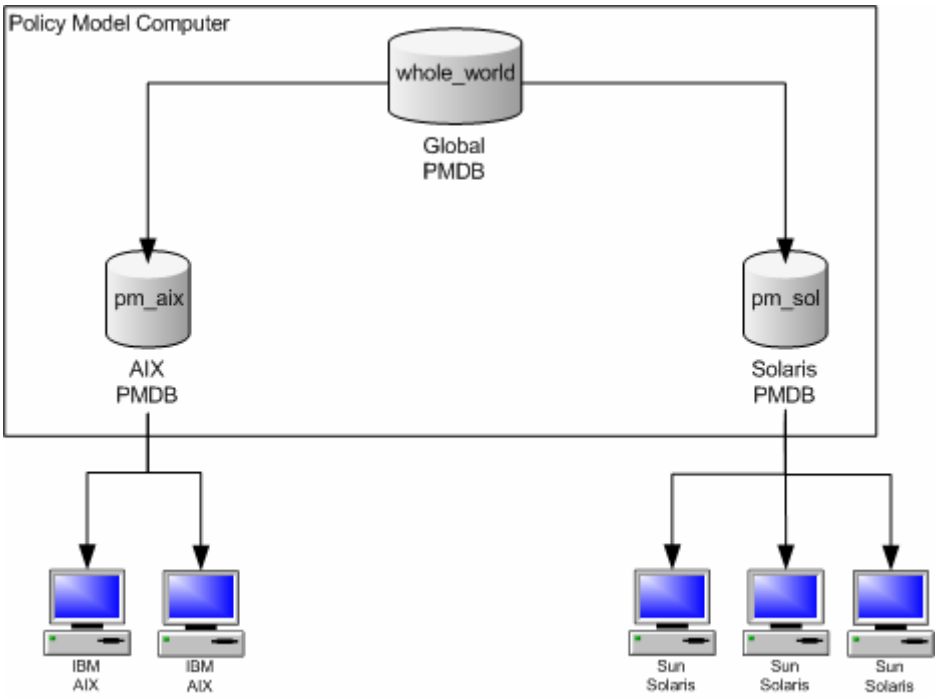
The following UNIX example also applies to a Windows architecture with small modifications.

In the example, the site consists of IBM AIX and Sun Solaris systems. Since the list of trusted programs on IBM AIX differs from the one on Sun Solaris, the PMDBs need to consider architecture dependency.

To set up a multiple-architecture PMDB, set up your PMDBs as follows:

1. Define a PMDB named whole_world, to contain the users, groups, and all other architecture independent policies.
2. Define a PMDB named pm_aix, to contain all the IBM AIX specific rules.
3. Define the PMDB pm_sol, to contain all the Sun Solaris specific rules.

   The PMDBs pm_aix and pm_solaris are subscribers of the PMDB whole_world. All IBM AIX computers at the site are subscribers of pm_aix. All Sun Solaris computers at the site are subscribers of pm_sol. The concept is illustrated in the following chart.



4. When you enter platform-independent commands in whole_world, such as adding a user or setting a SURROGATE rule, all databases at the site are automatically updated.
5. When you add a trusted program to pm_aix, only IBM AIX computers are updated, without affecting the Sun Solaris systems.

Methods for Centrally Managing Policies

eTrust AC lets you manage several databases from a single computer in two ways:

- Automatic rule-based policy updates

  Regular rules you define in a central database (PMDB) are automatically propagated to databases in a configured hierarchy.

  Note: Dual control is only available with this method and on UNIX only.

- Advanced policy management and reporting

  Policies (group of rules) you deploy are propagated to all databases in a configured hierarchy. You can also undeploy (remove) policies and report on deployment status, deployment deviation, and deployment hierarchy. You need to install and configure additional components to use this functionality.

Note: Advanced policy management and reporting offers additional functionality that is not available with automatic rule-based policy updates. However, to use advanced policy management you first need to configure your environment for automatic rule-based policy updates.

Automatic Rule-based Policy Updates

Single-rule policy updates (regular selang rules) you make in a central database are automatically propagated to the subscriber databases. By subscribing several computers to the same database, and by subscribing one database to another, you can create a hierarchy. You configure your environment for automatic rule-based policy updates after installation.

Note: This method of managing policies is limited to letting you make single-rule policy updates across your hierarchy. Other functionality is only available through implementing advanced policy management and reporting (see page 91).

How Automatic Rule-based Policy Updates Work

When you configure your environment for automatic rule-based policy updates, each rule you define in the central database is automatically propagated to all of its subscribers in the following way:

1. A rule is defined for any PMDB with at least one subscriber.
2. The PMDB sends the command to all subscriber databases.
3. The subscriber database applies the propagated command.
    a. If the subscriber database does not respond, the PMDB sends the command at a regular interval (by default, every 30 minutes) until the subscriber database has been updated.
    b. If a subscriber database is responding, but refuses to apply the command, the PMDB places the command in the Policy Model error log (see page 88).
4. If the subscriber database is a parent to other subscribers, it then sends the command to its subscribers.

Example: Removing a user from all computers in a hierarchy

If a user is deleted from a PMDB using the rmusr command, the same rmusr command is sent to all the subscriber databases. In this way, a single rmusr command can remove a user from many databases on a variety of computers.

How You Can Set Up a Hierarchy

eTrust AC uses the Policy Model service to propagate rule-based policy updates across the configured hierarchy. By subscribing several eTrust AC computers to the same PMDB, and by subscribing one PMDB to another, you create a hierarchy.

The most straightforward way to set up the PMDB hierarchy is as you install eTrust AC, so it is worth thinking through how you want to structure the hierarchy before you begin the installation. Make sure that all the hosts in the PMDB hierarchy are part of the same network, since the parent PMDB and its subscribers must be able to communicate with each other. That is, the parent must be able to connect with each of its subscribers by name, and every subscriber must be able to connect to the parent PMDB by name.

Note: For more information about installing eTrust AC, see the *Implementation Guide*.

If you want to change the configuration that you created during installation or if you did not create a PMDB structure during installation, you can change or create the PMDB configuration at any time. You can do this in one of the following ways:

- With Policy Manager
- With the sepmd utility

To create a PMDB hierarchy and enable automatic rule-based policy updates after installation, do the following:

1. Create and configure the master PMDB.
2. (Optional) Create and configure subscriber PMDBs.
3. Define parent PMDBs for the subscribing computers, called *end-points*.

Update Subscribers

When updating subscribers, the Policy Model performs the following actions:

1. The Policy Model tries to fully qualify subscriber names as they are added or deleted from the Policy Model.
2. The PMDB service, sepmdd, attempts to update a subscriber database.
3. If the maximum time elapses and the service does not succeed in updating a subscriber, it skips that particular subscriber and tries to update the remainder of the subscribers on its list.
4. After it completes its first scan of the subscriber list, sepmdd then performs a second scan, in which it tries to update the subscribers that it did not succeed in updating during its first scan.

Note: Whenever a PMDB encounters an error while propagating updates to subscribers, the sepmdd service creates an entry in the Policy Model error log file (see page 88). This file, ERROR_LOG, is located in the PMDB directory (see page 76).

Update a Policy Model Database

Working at the computer where the PMDB resides does not automatically update the PMDB itself. To update a PMDB, you need to specify it as your target database.

You can specify the PMDB using selang or Policy Manager. To specify a target database using selang, use the hosts command in the selang command shell:

```
eTrustAC> hosts <pmd_name>@<pmd_host>
```

All selang commands now update the policy model database specified. The commands then automatically propagate to the active databases on this computer and of all subscriber computers.

Note: For more information about Policy Manager, see the *Policy Manager online help*.

Example: Specify a target PMDB

To set the target database to policy1 on myPMD_host, use the following command:

```
eTrustAC> hosts policy1@myPMD_host
```

If you now enter the newusr command, the new user is added to the policy1 database as well as the active databases on this computer and of all subscriber computers.

Clean Up the Update File

The sepmd utility automatically writes each update it receives in the updates.dat file. To prevent the file from growing too large, we recommend that you delete processed updates from the file periodically.

To clean up the update file, use the following command:

```
<eTrustAC_InstallDir>/bin sepmd -t pmdbName auto
```

sepmd calculates the offset of the first update entry that has not been propagated and deletes all the update entries before it.

Propagate and Synchronize Passwords

Once you set up a PMDB hierarchy, you can use it to keep user passwords synchronized throughout your system when the user passwords are changed with the Windows User Manager or software other than eTrust AC.

Note: eTrust AC also supports mainframe password synchronization (see page 149).

To propagate and synchronize passwords

1. Create a PMDB Hierarchy.
2. Enter the name of the appropriate parent PMDB as the passwd_pmd entry value in the registry on every station on which users or administrators may change passwords.

   ```
   HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrustAccessControl\
   eTrustAccessControl\passwd_pmd
   ```

   The PMDB then propagates the password change to all its subscribers. If the passwd_pmd value is blank, eTrust AC checks the secondary_pmd value and sends new and updated passwords to the PMDB listed in this value, unless it also is blank.

Note: If the PMDB sends a user password to a subscriber in which the user is not defined, settings are not changed and the user remains undefined for the subscriber.

Remove a Subscriber

If you no longer want to propagate updates to a particular subscriber, you should remove it.

To remove a subscriber

1. Remove the computer from the subscription list:

   `sepmd -u <PMDB_name> <computer_name>`

   The computer is removed from the Policy Model subscription list.

2. Shut down seosd on the computer that you removed from the subscription list:

   `secons -s`

   The seosd service is shut down.

3. Delete the value of the parent_pmd registry value in the following registry key on the computer you removed from the subscription list:

   `HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrustAccessControl\eTrustAccessControl`

   The computer will stop accepting updates from the parent PMDB.

4. Restart seosd.

   The active database on the computer that you removed from the subscription list, is no longer a subscriber of the specified PMDB.

Note: Once the database is unsubscribed from the PMDB, the PMDB no longer sends commands.

Filter Updates

If you want your PMDB to update different subsets of data at different subscriber databases, you need to define which records are sent to subscriber databases.

To filter updates

1. Configure PMDBs to serve as parents to subsets of subscribers.
2. Modify the *Filter* registry entry in the registry key of the parent PMDB, to point to a filter file you set up on the same computer.

   Updates to the subscriber databases are then limited to the records that pass the filter.

Policy Model Filter File

A filter file consists of lines, each with six fields. The fields contain information on:

- The form of access permitted or denied.

  For example, EDIT or MODIFY

- The environment affected:

  eTrust, UNIX, or Native

- The class of the record.

  For example, USER or TERMINAL

- The objects, within the class, that the rule covers.

  For example, User1, AuditGroup, or COM2

- The properties that the record grants or cancels.

  For example, OWNER and FULL_NAME in the filter line means that any command having those properties is filtered. You must enter each property exactly as it appears in the *Reference Guide*.

- Whether such records should be forwarded to the subscriber database or not:

  PASS or NOPASS

The following rules apply to each line in the filter file:

- You can use an asterisk (*) to denote all possible values in any field.
- If more than one line covers the same records, the *first* applicable line is used.
- Spaces separate the fields.
- In fields with more than one value, semicolons separate the values.
- Lines beginning with **#** are considered a comment line.
- Empty lines are not allowed.

Example: Filter file

The following example describes a line from a filter file:

CREATE	**eTrust**	**USER**	*****	**FULL_NAME;OBJ_TYPE**	**NOPASS**
↑	↑	↑	↑	↑	↑
form of access	environment	class	record name (* =all)	properties	treatment

In this example, if we name the file with this line Printer1_Filter.flt and edit the registry for PMDB PM-1 so that filter=C:\Program Files\CA\eTrustAccessControl\Printer1_Filter.flt, then PMDB PM-1 will not propagate to its subscribers any records that create new users with the FULL_NAME and OBJ_TYPE property.

Policy Model Error Log File

The Policy Model error log, which is organized chronologically, looks similar to this:

Error Text	Error Category
20 Nov 03 11:56:07 (pmdb1): fargo nu u5 0 Retry ERROR: Login procedure failed (10068) ERROR: Cannot accept update from a non-parent PMDB (pmdb1@name.company.com) (10104)	Configuration Errors
20 Nov 03 19:53:17 (pmdb1): fargo nu u5 0 Retry ERROR: Connection failed (10071) Host is unreachable (12296)	Connection Errors
20 Nov 03 11:57:06 (pmdb1): fargo nu u5 560 Cont ERROR: Failed to create USER u5 (10028) Already exists (-9) 20 Nov 03 11:57:06 (pmdb1): fargo nu u5 1120 Cont ERROR: Failed to create USER u5 (10028) Already exists (-9)	Database Update Errors

The Policy Model error log is in binary format; you can view it only by entering the following command:

<eTrustAC_InstallDir>/bin sepmd -e *pmdname*

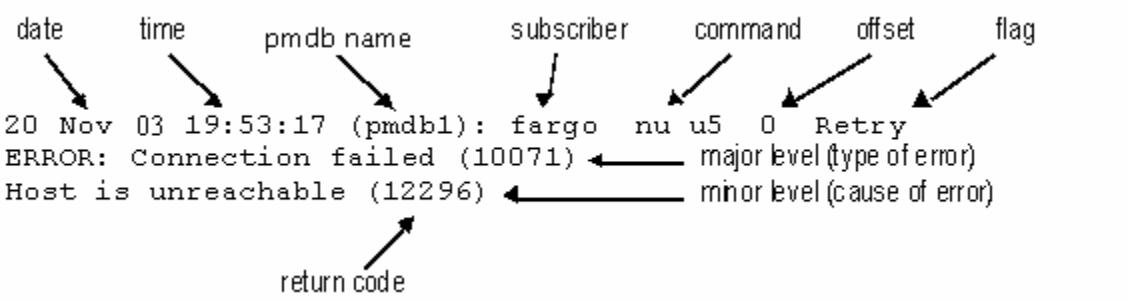
Note: Do not manually delete an error log (for example, with the UNIX rm command). To delete the log, only use the following command:

<eTrustAC_InstallDir>/bin sepmd -c *pmdname*

Important! The error log in eTrust AC r5.1 and later versions has a format that is not compatible with the format of earlier versions. sepmd cannot handle error logs from these earlier versions. When you upgrade to a version that has this format, the old error log is copied to ERROR_LOG.bak; a new log file is created when you start sepmdd.

Example: PMDB update error message

The following example shows a typical error message:



- The top line always consists of the date, time, and subscriber. The command that generated the error appears next, followed by the offset (in decimal format), which indicates the location of the failed update inside the updates file. Lastly, the flag indicates whether the PMDB retries the update automatically or continues without it.
- The second line shows an example of a major level message (what type of error occurred) and its return code.
- The third line displays an example of a minor level message (why the error occurred), and its return code.

Example: Error message

A command may generate and display more than one error. Also, an error may consist of a major level message, a minor level message, or both.

The following error has only one message level:

```
Fri Dec 29 10:30:43 2003 CIMV_PROD:Release failed. Return code = 9241
```

This message occurs when sepmd pull attempts to release a subscriber that is already available.

Native Policy Model Repositories

You can store all native environment user and group object types in the PMDB. By storing this information in the PMDB, you can receive information about objects using show commands (such as show user or show group). The returned objects are an image of the actual objects that are defined in Windows or UNIX subscribers.

After connecting to a Policy Model, a user can choose the following environments:

- eTrust
- Native
- NT
- UNIX

Note: Native operates exactly as Windows while you are working in a Windows operating system, or exactly as UNIX while you are working on a UNIX operating system.

To use native environment repositories, use the following commands:

- Enter the following commands at the selang prompt:

  ```
  env NT; find
  ```

  Your results list all the native environment object types.

  Note: For descriptions of these object types, see the Windows environment classes and properties in the *Reference Guide*.

- Enter the following commands to receive a list of NT and Active Directory USER properties:

  ```
  env NT; ruler user
  ```

- Enter the following commands to receive a list of NT and Active Directory GROUP properties:

  ```
  env NT; ruler group
  ```

If a Policy Model is a subscriber of another (parent) Policy Model, it receives data from a parent through propagation and saves in the database all user and group properties, so you can see them and change them.

Note: For more information, see the sepmd utility in the *Reference Guide*.

Advanced Policy Management and Reporting

Multiple-rule policies (script files) you create can be stored and then deployed in a configured hierarchy. Using this policy-based method, you can store policy versions and then deploy and undeploy those. You can also create reports on deployment status, deployment deviation, and deployment hierarchy.

Note: Dual control is *not* available with this method and is only available on UNIX.

Environment Architecture

To use advanced policy management and reporting, you need to install and configure the following additional components:

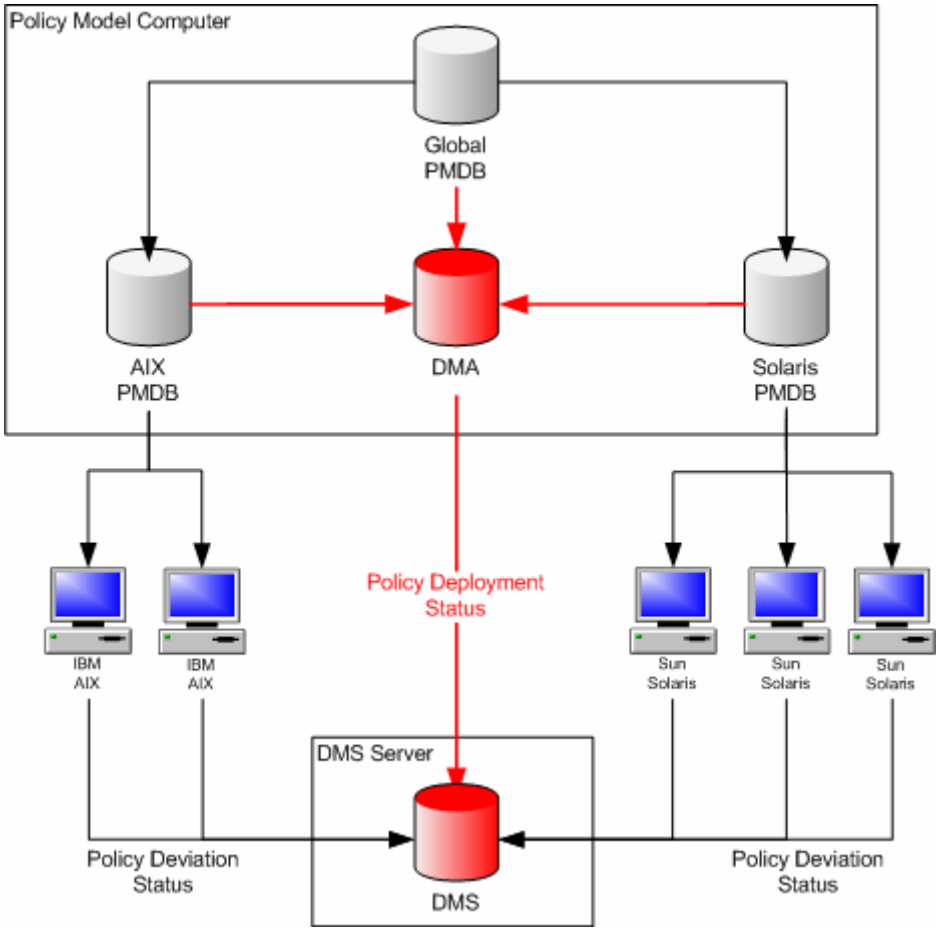
- A Deployment Map Server (DMS) (see page 93) on a central computer that is designated for this purpose.
- A Deployment Map Agent (DMA) (see page 93) on each computer which has at least one PMDB.

Note: Advanced policy management and reporting requires setting your environment for automatic rule-based policy updates. After installing the DMS and DMA on appropriate computers, configure your parent and subscriber databases (see page 82).

Example: A two-tiered hierarchy with a central DMS

Note: The following UNIX example also applies to a Windows architecture with small modifications.

In this example, the site consists of IBM AIX and Sun Solaris systems. Since the list of trusted programs on IBM AIX differs from the one on Sun Solaris, the PMDBs need to consider architecture dependency. For management and reporting purposes, we will set up the DMS and DMA to support the environment we created when we configured a multiple-PMDB environment (see page 78). The DMS stores all policy deployment and deviation information and eTrust AC lets you create reports from this information.



Deployment Map Server (DMS)

The DMS sits at the core of advanced policy management and reporting. The purpose of DMS is to keep an up-to-date map of the eTrust AC deployment hierarchy and the status of policies deployed on each computer. Having this data in a central location (rather than connecting to each database in a hierarchical manner) reduces the time required for generating reports. In addition, the DMS stores versions of your policies that you can later deploy and undeploy as required.

A DMS is a PMD node and it uses a PMDB as its data repository. It collects the data it receives from notifications from each PMD node it is configured for.

Deployment Map Agent (DMA)

The DMA is an agent that is responsible for PMD communication with the DMS. You should install a DMA on each PMD node (at least one PMDB). The DMA on the parent PMDB is responsible for notifying the DMS of policy deployment status and hierarchy changes. End-points send only policy deviation status directly to the DMS.

Note: Do not install a DMA on end-point computers.

The DMA uses standard eTrust AC communication and encryption mechanisms for communication with the DMS.

Note: You do not need to define the DMA as the parent of the DMS to enable DMA\DMS communication.

Advanced Policy Management Classes

In order for the DMS to keep an up-to-date map of the eTrust AC deployment hierarchy and the status of policies deployed on each computer, it uses specific eTrust AC classes.

HNODE

Each HNODE object represents a node in a hierarchy. It holds information about the particular node it represents, its subscribers, and the parent PMDBs. In addition, each HNODE object holds information about the policies that should be deployed on the node it represents, and the status of each policy (deployed, deployed with errors, and so on)

The name of the HNODE object depends on the type of node it represents:

- The actual host name for end-points.

  For example, myhost.mydomain.com

- The Policy Model name for a PMDB.

  For example, mypmd@hostB.domain.com

POLICY

Each POLICY object represents a version of a policy that may be deployed on any part of the HNODE hierarchy. It contains information about where the associated policy scripts are stored (in which RULESET object) and which nodes it should be deployed on.

The name of the object is the name of the policy, suffixed by a version number (*policy_name#xx*).

RULESET

Each RULESET object holds both the deployment and undeployment (removal) scripts that are associated with a policy version.

The name of the object is based on the respective POLICY object name.

Note: For more information about these classes, see the *Reference Guide*.

How You Set Up a Hierarchy for Advanced Policy-based Management and Reporting

eTrust AC uses a DMS to keep an up-to-date map of the eTrust AC deployment hierarchy and the status of policies deployed on each computer. By installing and configuring the appropriate components on each computer in your hierarchy, you enable policy-based management and reporting.

To enable policy-based management and reporting, do the following:

1. Install a DMS on a central computer.

   A DMS can be installed during eTrust AC installation or by the dmsmgr utility.

2. Install a DMA on each PMD node.

   A DMA can be installed during eTrust AC installation or by the dmsmgr utility.

3. Install advanced policy management and reporting functionality on each eTrust AC computer.

   This configures the deviation calculation for sending policy deviation status to the DMS.

4. Set up a hierarchy (see page 82).

   As you set up your hierarchy, HNODE objects representing each node in your hierarchy are added in the DMS.

   Important! When you uninstall eTrust AC from a computer or delete a PMDB, the HNODE object remains. You need to remove obsolete object nodes (see page 97). If you unsubscribe databases from the hierarchy, the HNODE object remains but its link to the parent node is removed. You do not need to remove this object but it will maintain links to policy objects that were previously deployed on that node.

Note: For information about installing a DMS, a DMA, and configuring advanced policy management and reporting functionality, see the *Implementation Guide*.

DMS Notifications

When you configure your environment for advanced policy management and reporting, components in your hierarchy notify the DMS of status changes in three areas:

- Hierarchy change.

  A notification is sent when a subscriber (PMD node or end-point) is added or deleted.

- Policy deployment and undeployment.

  A notification is sent when a policy is being deployed or undeployed on a subscriber (PMD node or end-point). The policy details and the deployment status are then updated according to the result of the operation (succeeded, failed, and so on).

- Deviation status.

  A notification is sent when an eTrust AC end-point calculates policy deviation and sends the result (deviation found or not found).

Note: Hierarchy change and policy deployment and undeployment notifications are sent by the DMA from PMD nodes only. Deviation status notifications are sent by the deviation calculator from eTrust AC end-points only.

How Hierarchy and Policy Status Notifications Work

The DMA sends hierarchy changes and policy status notifications to the DMS. DMA notifications are handled in the following way:

1. The DMA stores notification messages in an update file.

   These are hierarchy change and policy deployment and undeployment notifications.

2. The DMA contacts the DMS:

   - If a DMS is unavailable, the DMA tries to communicate with the DMS periodically, until all messages are successfully sent.
   - If the DMS is available, the DMA sends the stored notifications.

   Note: Each DMA communicates directly with a DMS, bypassing the hierarchy and reducing dependency.

3. The DMS stores the information it receives from each DMA for later use.

   Each time you create a report, eTrust AC retrieves the information on the DMS.

How Deviation Notifications Work

The deviation calculator is installed with eTrust AC and runs locally on the end point it calculates deviations for. The deviation calculator performs the following actions and sends these notifications to the DMS:

1. The calculation process is triggered by sending a selang command (start devcalc) to the end point.

   We recommend you do this as a scheduled execution by a customized script.

2. Once the calculation is complete, the deviation calculator stores the result in a data file.

   The file is <*eTrustAC_Dir*>\data\devcalc\deviation.dat

   Note: The reporting utility can retrieve the deviation details (using the -dev option). Alternatively, you can retrieve the contents of the data file using the *get devcalc* command on the end-point.

3. The deviation calculator then sends the deviation status (was a deviation found or not) to the DMS.

   The deviation itself (the contents of the data file) is not sent with the status notification.

Remove Obsolete Nodes from a Hierarchy

The DMS stores information about your hierarchy. If you remove a computer from the hierarchy when you uninstall eTrust AC from that computer, the DMS still contains a reference to that node. As a routine maintenance procedure, you should clean the DMS from these obsolete nodes.

To remove obsolete nodes from a hierarchy run the dmsmgr utility on the DMS computer to perform a routine clean up:

```
dmsmgr -dms -cleanup <number_of_days>
```

where <*number_of_days*> is the minimum number of days in which the eTrust AC node has been unavailable for.

Note: You can also manually delete a specific node by issuing the following selang command on the DMS computer:

```
rr HNODE <HNODE_name>
```

How Advanced Policy-based Management Works

Advanced policy-based management lets you store, deploy, and undeploy policy versions, and later create reports on deployment status, deployment deviation, and deployment hierarchy. Each policy is a pair of selang script files you create. The first script file is called a *deployment script* and contains a set of selang commands that construct the policy. The second script file is an *undeployment script* and contains commands that are required for undeploying (removing) the policy from the end-point database.

Each policy is applied in two stages to target databases you specify:

1. You store policy details in the DMS.

   Policy details include the deployment and undeployment scripts, and the policy signature which is created automatically (used to detect variations of the same policy).

   If policy details cannot be stored in the DMS, make sure you:

   - Store the policy from a computer that has TERMINAL rights to the DMS.
   - Have sub-administration permissions to the POLICY and RULESET classes on the DMS.
   - Do not have a deployment or undeployment script that contains a syntax error.

2. The utility stores the policy using automatic version-control.

   Depending on whether the policy already exists on the DMS, the utility does *one* of the following:

   - If the policy name does not exist on the DMS, it creates the first version of the policy (*policy_name*#01).
   - If the policy name already exists on he DMS, it creates a new policy version, incrementing the highest found policy version by one.

3. You deploy a stored policy version to target databases.

   If the stored policy cannot be deployed in your target hierarchy, make sure you:

   - Deploy from a computer that has TERMINAL rights to your target Policy Model root.
   - Have sub-administration permissions to the POLICY, RULESET, and HNODE classes on the DMS and each database in the hierarchy where you are deploying the policy.
   - Have sub-administration permission on the target Policy Model root computer.
   - Do not have a version of the same policy already deployed on a host which is part of the deployment hierarchy.

4. Each rule-the selang commands specified in the deployment script-is executed on the target databases.

   If a rule cannot be deployed in a certain database, the policy is considered as deployed with failures (status *Failed*).

   This happens if you try to deploy a policy on a host and the deployment script contains:

   - A reference to an object that does not exists. For example:

     ```
     cr FILE /does_not_exist comment(123)
     ```

     For this reason, the policy deployment script must be self-sufficient. That is, the deployment script must be constructed so that it creates all the resources it uses.

   - A command that causes an error.

   - A command you do not have sub-administration permissions to execute.

5. The policy status is recorded.

   Policy status can be Deployed, Undeployed, Transferred, Failed (deployed with failures), Queued, TransferFailed, SigFailed (signature failed), UndeployFailed (undeployed with failures), or Unknown.

   Note: If a policy deployed with errors, you need to view the log file on the computer where the policy was deployed with errors.

Once the policy details are stored on the DMS and then deployed on the target databases, if a target database is a PMDB, the policy is propagated throughout the hierarchy using the automatic rule-based policy updates mechanism.

When a new subscriber is added to the hierarchy, all policies are propagated through the hierarchy and the DMS is notified that a node was added to the hierarchy.

Administration Requirements

You can run the policy deployment utility (policydeploy) from any computer, as long as eTrust AC is installed. In order to store policies on the DMS, or deploy and undeploy policies on databases in your hierarchy, you and the computer you are working from need to have appropriate permissions.

To store policies on the DMS:

- The *computer* you are running the policydeploy utility from, needs to have terminal rights (TERMINAL class) for the DMS.
- *You* need to have sub-administration rights for the POLICY and RULESET classes on the DMS.

To deploy and undeploy policies throughout your hierarchy:

- The *computer* you are running the policydeploy utility from, needs to have terminal rights (TERMINAL class) for the computer which is the target Policy Model root.
- *You* need to have:
  - Read rights for the POLICY and RULESET classes, and sub-administration rights for the HNODE class on the DMS.
  - Sub-administration rights for the POLICY, HNODE, and RULESET classes on each database in the hierarchy where you are deploying the policy.
  - Appropriate sub-administration rights on each of the databases in the hierarchy where you are deploying the policy.

    These are the permissions necessary to deploy the selang commands that form the policy on each of these computers.

    For example, you'll need sub-administration rights for the FILE class if you are creating a new file resource:

    ```
    nr FILE /inetpub/* defaccess(none)
    ```

How to Deploy Approved Policy Versions

Using advanced policy-based management you can store a draft version of a policy, have it reviewed and modified as required, then deploy the approved version.

To deploy approved policy versions, do the following:

1. Store a policy version on the DMS.

   Once you have a stored policy version, the policy can be reviewed and deployed.

2. Review the policy (see page 104).

   Anyone with read permissions to the POLICY and RULESET objects can view the policy and its associated rules.

3. If required, store a new version of the policy with the approved changes.

   Whenever you need to update a policy, you must store a new version of the policy that contains the required modified policy deployment and undeployment rules.

4. Deploy the approved policy version to your hierarchy (see page 105).

Store a Policy Version

Every policy you store on the DMS automatically gets a version number. The first time you store a policy it receives a version number "01". For example, the first time you store policy *myPolicy*, the policydeploy utility creates a POLICY object named *myPolicy#01*. Every time you store a policy that already exists on the DMS, the latest stored version of the policy will be incremented by one to create the new policy version. For example, when you store a version of *myPolicy* for the twenty-eighth time, the policydeploy utility creates a POLICY object named *myPolicy#28*.

You can then view stored policies and deploy them to your hierarchy as required.

To store a policy version

1. Create a new script file with selang deployment commands.

   These are the commands necessary to construct the policy you want to deploy on each computer in a hierarchy.

   Important! Policy deployment does not support commands that set user passwords. Do not include such commands in your deployment script file. Windows (native) selang commands are supported but will not show in deviation reports.

2. Create a new script file with selang undeployment commands.

   These are the commands necessary for undeploying (removing) the policy from a computer in the hierarchy.

   Note: These commands are used by default when you undeploy a policy from a targeted hierarchy unless you supply a new policy undeployment script at the time you undeploy a policy.

3. Run the policydeploy utility with the -store option:

   `policydeploy -store name -ds file1 -uds file2 [-dms list]`

   where *name* is the name of the policy you want to store, *file1* is the full path and name of your deployment script file, *file2* is the full path and name of your undeployment script file, and *list* is an optional comma-separated list of DMS nodes.

   The policydeploy utility prompts you for whether to create a new version of the policy on the DMS.

   Note: Policy *names* cannot include the # (hash) character which is reserved for denoting policy version numbers and is added automatically.

4. Enter y to confirm this action.

   The policydeploy utility creates a new version of the policy on the DMS.

Example: Storing an IIS 5 protection policy

The following example shows you how to store a policy for securing Internet Information Services (IIS) 5 web servers. This is the first time we store this policy on the DMS.

We will deploy the policy IIS5 on a Policy Model hierarchy for which iis5@host.company.com is the root PMDB.

1. Save a file called IIS5.selang with the following IIS script:

   ```
   nu inet_pers owner(nobody)
   nr FILE c:\InetPub\wwwroot\* defaccess(none) owner(nobody)
   authorize FILE c:\InetPub\wwwroot\* uid(inet_pers) access(all)
   nr FILE c:\InetPub\wwwroot\scripts defaccess(none) owner(nobody)
   nr FILE *.asp defaccess(none) owner(nobody)
   authorize FILE *.asp uid(inet_pers) via(pgm(inetinfo.exe)) access(read,
   execute)
   ```

   These are the commands necessary to deploy an IIS 5 protection policy.

2. Save a file called IIS5_rm.selang with the following script:

   ```
   ru inet_pers
   rr FILE c:\InetPub\wwwroot\*
   rr FILE c:\InetPub\wwwroot\scripts
   rr FILE *.asp
   ```

   These are the commands necessary to undeploy the IIS 5 protection policy we created in step 1.

3. Run the policydeploy utility:

   ```
   policydeploy -store IIS5 -ds IIS5.selang -uds IIS5_rm.selang
   ```

   This stores the first version of the IIS5 policy (IIS5#01) as defined in IIS5.selang and IIS5_rm.selang on the DMS.

View the Rules Associated with a Policy

Once a policy is stored on the DMS, anyone with read permissions to the POLICY and RULESET objects can view the policy and its associated rules. If you do not know which is the latest version of the stored policy, you can find this out first.

To view the rules associated with a policy

1. Connect to the DMS via selang:

   `hosts dms_name@hostname`

   You can now query our DMS via selang.

2. If you want to know which is the latest version of the policy, issue the following selang command to find all versions of the policy:

   `find POLICY policy_name#*`

   The selang window lists all versions of the *policy_name* policy.

3. Issue the following selang command to view the policy deployment and undeployment scripts:

   `sr RULESET policy_name#xx`

   where *xx* is the number of the policy you want to view the rules for.

   The selang window displays the *policy_name#xx* RULESET object, including the deployment and undeployment rules that relate to the *xx* version of the *policy_name* policy.

Deploy a Stored Policy Version

You can deploy a stored version of a multiple-rule policy in your hierarchy in a way that lets you undeploy the policy later on, and create reports on deployment status, deployment deviation, and deployment hierarchy.

To deploy a stored policy version

Do *one* of the following:

- If you want to deploy the *latest* version of the stored policy, run the policydeploy utility specifying the policy name and target hierarchy:

  ```
  policydeploy -deploy name -root db1[,db2] [-dms list]
  ```

  The utility finds the latest version of the policy on the DMS with the name you supplied, and tries to deploy it on the target databases. Policy commands are then propagated to subscribing databases if any exist.

- If you want to deploy a *specific* version of the stored policy, run the policydeploy utility specifying the policy name, its version, and a target hierarchy:

  ```
  policydeploy -deploy name#xx -root db1[,db2] [-dms list]
  ```

  The utility tries to deploy the specified version of the policy on the target databases. Policy commands are then propagated to subscribing databases (if any exist).

Note: For more information about the policydeploy utility, see the *Utilities Guide* for UNIX or the *Reference Guide* for Windows.

Important! You cannot deploy a policy if a version of the same policy is already deployed on any of the hosts within the deployment hierarchy.

Example: Deploying an IIS 5 protection policy

The following example shows you how to deploy a policy for securing Internet Information Services (IIS) 5 web servers. We will review the fourth version of policy IIS5 and then deploy it on a Policy Model hierarchy for which iis5@host1.company.com is the root PMDB. Policy IIS5 is stored on the crDMS@cr_host.company.com DMS node.

1. Connect to the DMS via selang:

   ```
   hosts crDMS@cr_host.company.com
   ```

   You can now query our DMS via selang.

2. If you're not sure what is the latest available version of the policy, issue the following selang command to find all versions of the policy:

   ```
   find POLICY IIS5#*
   ```

   The selang window lists all versions of the IIS5 policy.

3. Issue the following selang command to view the policy deployment and undeployment scripts:

   ```
   sr RULESET IIS5#04
   ```

   The selang window displays the IIS5#04 RULESET object, including the deployment and undeployment rules that relate to the fourth version of the IIS5 policy.

4. In a command line window, run the policydeploy utility:

   ```
   policydeploy -deploy IIS5#04 -root iis5@host1.company.com
   ```

   This deploys the fourth version of the IIS5 policy on the PMD hierarchy under iis5@host.company.com

Undeploy a Policy

You can undeploy multiple-rule policies from a targeted hierarchy if you no longer want to have the policy deployed on those computers. You also need to undeploy a policy if you want to modify it (create an updated version of the policy).

To undeploy policy

1. (Optional) Create a new script file with selang undeployment commands.

   These are the commands necessary to undeploy (remove) the policy from a computer in the hierarchy.

   If you do not create and specify a new undeployment script, the undeploy command uses the script assigned to this policy when it was deployed.

   Important! Even if you specify a policy undeployment script, the DMS still records the original rules that were provided when you stored the policy and not the new script which is used to undeploy the policy.

2. Do *one* of the following:

   – If you want to undeploy the *latest* version of the policy, run the policydeploy utility specifying the policy name and target hierarchy:

     `policydeploy -undeploy name -root db1[,db2] [-dms list] [-uds file2]`

     The utility finds the latest version of the policy on the DMS with the name you supplied and tries to undeploy it from the target databases. Policy undeployment commands are then propagated to subscribing databases if any exist.

     Important! If the policy version on any end-point in your hierarchy contains other versions than the latest version found on the DMS, you will have to undeploy each of these specific versions explicitly.

   ▪ If you want to undeploy a *specific* version of the policy, run the policydeploy utility specifying the policy name, its version, and a target hierarchy:

     `policydeploy -undeploy name#xx -root db1[,db2] [-dms list] [-uds file2]`

     The utility tries to undeploy the specified version (*xx*) of the policy from the target databases. Policy undeployment commands are then propagated to subscribing databases (if any exist).

   Note: For more information on the policydeploy utility, see the *Utilities Guide* for UNIX or the *Reference Guide* for Windows.

Note: When you undeploy a policy, the DMS reports that the status of the policy is *Undeployed*. The POLICY and RULESET objects remain on all of the hosts the policy version was deployed on (including the DMS) so that it can be redeployed or queried at a later time.

Modify a Deployed Policy

To modify a deployed policy, you need to first undeploy the deployed policy version, store a new version of the policy with the modified deployment and undeployment scripts, and then redeploy the policy using the new version.

To modify a deployed policy

1. Store a new policy version.

   A new version of the policy is stored on the DMS.

2. Undeploy the policy (see page 107).

   The policy is undeployed from the target hierarchy.

3. Deploy the new version of the policy (see page 105).

   The policy is redeployed to the target hierarchy with the modified policy.

How Advanced Policy Reporting Works

Advanced policy reporting lets you create reports on deployment status, deployment deviation, and deployment hierarchy for a configured hierarchy and for policies that were created using the advanced policy-based management method. The report generation utility (policyreport) generates point-in-time (static) HTML reports based on DMS contents.

The policyreport utility creates hierarchy and policy reports by performing the following actions:

1. The utility queries the DMS for the requested information.

   The information retrieved depends on the type of report generated.

2. The utility queries end points for policy deviation results, if a deviation calculation is requested.

   The deviation status exists on the DMS, but the actual deviation needs to be retrieved from each end-point.

3. The utility generates a set of XML documents.

   This is an XML report.

4. The utility formats the XML report into HTML.

   The report is now ready for viewing in a browser.

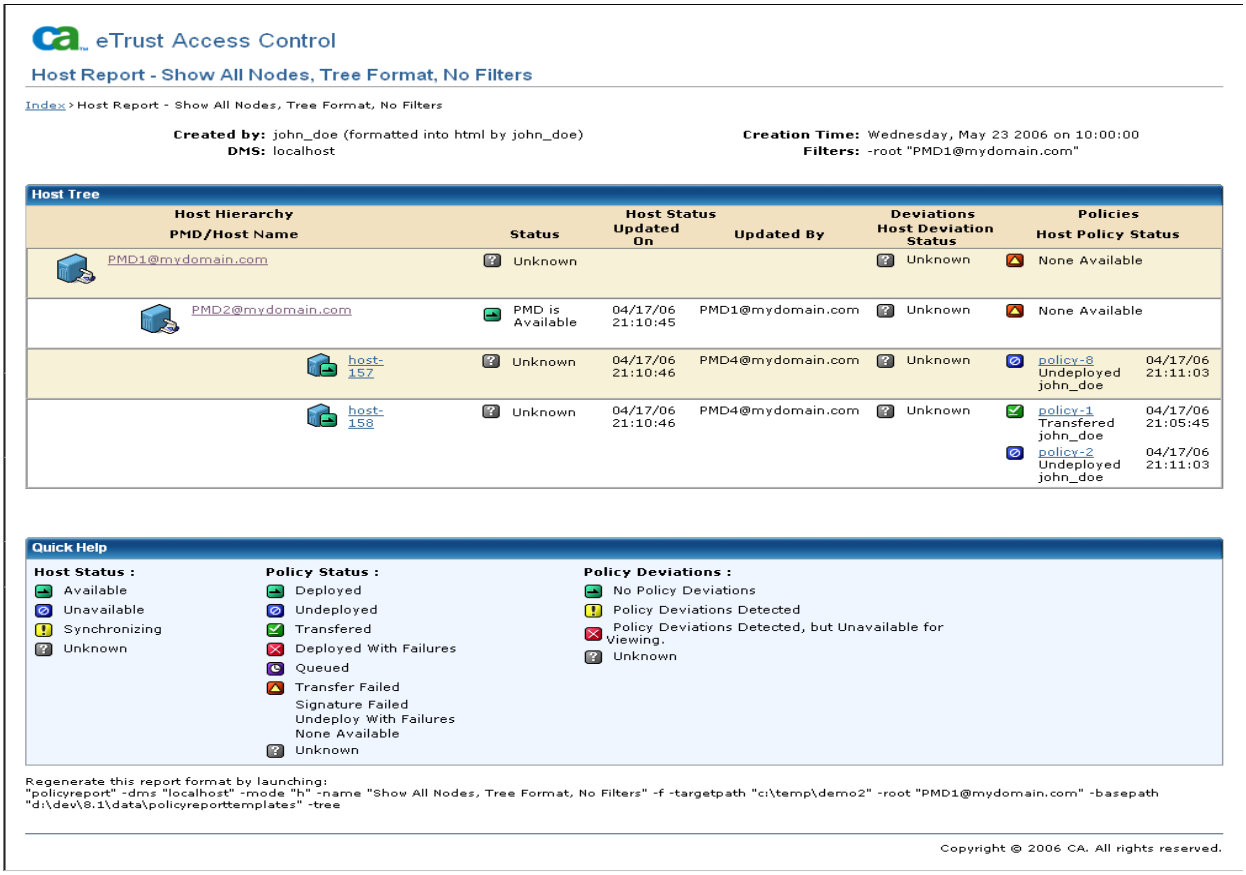
Note: The policyreport utility stores the report in a subdirectory you specify using the -name option under the directory you specify using the -targetpath option.

Report Types

The report generation utility lets you view the policies deployed across your hierarchy in two modes:

- By host

  Host reports present computer-centric information. Use this mode when you want to view your environment by host. In this mode you can see how your computers are configured, what is the status of each computer in your hierarchy, which computers have which policies and what their status is, and how the actual rules deployed deviate from the rules that should be deployed on each computer.



eTrust Access Control

Host Report - Show All Nodes, Tree Format, No Filters

Index > Host Report - Show All Nodes, Tree Format, No Filters

Created by: john_doe (formatted into html by john_doe)
DMS: localhost
Creation Time: Wednesday, May 23 2006 on 10:00:00
Filters: -root "PMD1@mydomain.com"

Host Tree

Host Hierarchy PMD/Host Name	Status	Host Status Updated On	Updated By	Deviations Host Deviation Status	Policies Host Policy Status
PMD1@mydomain.com	Unknown			Unknown	None Available
PMD2@mydomain.com	PMD is Available	04/17/06 21:10:45	PMD1@mydomain.com	Unknown	None Available
host-157	Unknown	04/17/06 21:10:46	PMD4@mydomain.com	Unknown	policy-8 Undeployed john_doe 04/17/06 21:11:03
host-158	Unknown	04/17/06 21:10:46	PMD4@mydomain.com	Unknown	policy-1 Transfered john_doe 04/17/06 21:05:45 policy-2 Undeployed john_doe 04/17/06 21:11:03

Quick Help

Host Status :
- Available
- Unavailable
- Synchronizing
- Unknown

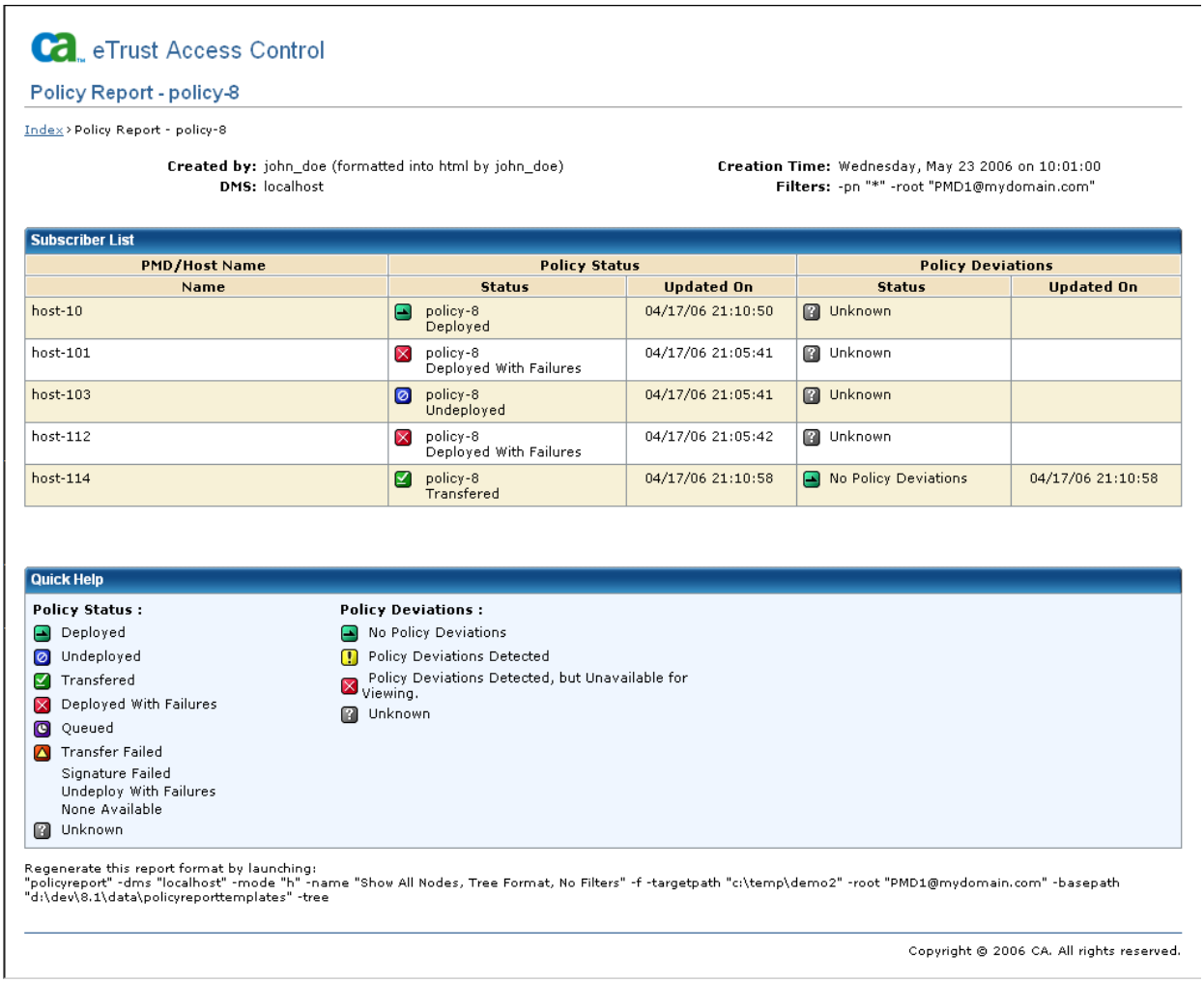
Policy Status :
- Deployed
- Undeployed
- Transfered
- Deployed With Failures
- Queued
- Transfer Failed
- Signature Failed
- Undeploy With Failures
- None Available
- Unknown

Policy Deviations :
- No Policy Deviations
- Policy Deviations Detected
- Policy Deviations Detected, but Unavailable for Viewing.
- Unknown

Regenerate this report format by launching:
"policyreport" -dms "localhost" -mode "h" -name "Show All Nodes, Tree Format, No Filters" -f -targetpath "c:\temp\demo2" -root "PMD1@mydomain.com" -basepath "d:\dev\8.1\data\policyreporttemplates" -tree

- By policy

  Policy reports present the policy-centric information. Use this mode when you want to view the status of one or more policies throughout your environment.



eTrust Access Control

Policy Report - policy-8

Index > Policy Report - policy-8

Created by: john_doe (formatted into html by john_doe)
DMS: localhost
Creation Time: Wednesday, May 23 2006 on 10:01:00
Filters: -pn "*" -root "PMD1@mydomain.com"

Subscriber List

PMD/Host Name	Policy Status		Policy Deviations	
Name	Status	Updated On	Status	Updated On
host-10	policy-8 Deployed	04/17/06 21:10:50	Unknown	
host-101	policy-8 Deployed With Failures	04/17/06 21:05:41	Unknown	
host-103	policy-8 Undeployed	04/17/06 21:05:41	Unknown	
host-112	policy-8 Deployed With Failures	04/17/06 21:05:42	Unknown	
host-114	policy-8 Transfered	04/17/06 21:10:58	No Policy Deviations	04/17/06 21:10:58

Quick Help

Policy Status :
Deployed
Undeployed
Transfered
Deployed With Failures
Queued
Transfer Failed
Signature Failed
Undeploy With Failures
None Available
Unknown

Policy Deviations :
No Policy Deviations
Policy Deviations Detected
Policy Deviations Detected, but Unavailable for Viewing.
Unknown

Regenerate this report format by launching:
"policyreport" -dms "localhost" -mode "h" -name "Show All Nodes, Tree Format, No Filters" -f -targetpath "c:\temp\demo2" -root "PMD1@mydomain.com" -basepath "d:\dev\8.1\data\policyreporttemplates" -tree

In addition to the type of report, you can also affect the output by:

- Choosing the part of the hierarchy for which you want the report generated.
- Generating the report for a single computer.
- Filtering by host name, status, or status update time, or by policy name or status (wildcards are supported).
- Including or excluding deviation calculation results.
- Selecting a tree-like format.
- Hiding report columns.

Create a Host Report

A host report lets you see how your computers are configured in the hierarchy, what is the status of each computer in your hierarchy, or which computers have which policies and what their status is.

To create a host report, run the policyreport utility in the *h* mode:

```
policyreport -name <name> -mode h -dms <dms_name> -root <pmd1>[,<pmd2>] -tree \
-targetpath <path>
```

Note: You can also fine-tune the report by using additional optional flags. For more information about the policyreport utility, see the *Reference Guide*.

Example: Create a report for computers whose host name matches a specified mask

The following example shows how you can use the policyreport utility to perform the following task:

- Generate a host report in the following directory:

  C:\eac_data\reports\production_March2006

- Retrieve the information is from the DMS:

  mainDMS on the **mainhost.domain.com** computer.

- Include only computers that are hierarchically beneath the following PMDB:

  rootPMD@root.domain.com

- Include only computers whose host name begins with:

  prod

```
policyreport -name production_March2006 -mode h \
-dms mainDMS@mainhost.domain.com -root rootPMD@root.domain.com -hn prod* \
-targetpath C:\eac_data\reports
```

The policyreport utility stores the report in a subdirectory we specified using the (-name production_March2006) option under the directory we specified using the -targetpath option (C:\eac_data\reports). We can update this report at a later time by adding the -f option that creates the report even if one already exists in the output directory.

Note: If you also specify the **-tree** flag, the report will show a graphical representation of the hierarchy. This includes parents of all computers in the report, even if the parents' host name does not match the specified mask.

Example: Create a report for computers whose status last changed within a certain date range

The following example shows how you can use the policyreport utility to perform the following task:

- Generate a host report in the following directory:

  C:\eac_data\reports\Feb06-Mar06

- Retrieve the information is from the DMS:

  mainDMS on the **mainhost.domain.com** computer.

- Include only computers that are hierarchically beneath the following PMDB:

  rootPMD@root.domain.com

- Include only computers whose host status was last updated in February 2006.

```
policyreport -name Feb06-Mar06 -mode h \
-dms mainDMS@mainhost.domain.com -root rootPMD@root.domain.com \
-sd 01-02-2006 -ed 28-02-2006 -targetpath C:\eac_data\reports
```

Example: Create a report that recalculates policy deviation results and store it in the current working directory

The following example shows how you can use the policyreport utility to perform the following task:

- Generate a host report in the following directory:

  *<working_directory>***/hierarchy_20March06**

- Retrieve the information is from the DMS:

  mainDMS on the **mainhost.domain.com** computer.

- Include only computers that are hierarchically beneath the following PMDB:

  rootPMD@root.domain.com

- Include deviation calculation results.

- Graphically represents the hierarchy using indentation.

```
policyreport -name hierarchy_20March06 -mode h -dms mainDMS@mainhost.domain.com \
-root rootPMD@root.domain.com -targetpath -dev -tree
```

Create a Policy Report

A policy report lets you see which policies are deployed on which computers.

To create a policy report, run the policyreport utility in the *p* mode:

```
policyreport -name <name> -mode p -dms <dms_name> -root <pmd1>[,<pmd2>] \
-targetpath <path>
```

Note: You can also fine-tune the report by using additional optional flags. For more information about the policyreport utility, see the *Reference Guide*.

Example: Create a report for all versions of a specified policy

The following example shows how you can use the policyreport utility to perform the following task:

- Generate a policy report in the following directory:

  C:\eac_data\reports\iis5Policies_March2006

- Retrieve the information from the following DMS:

  mainDMS on the **mainhost.domain.com** computer.

- Include only computers (subscribers) that are hierarchically beneath the following PMDB:

  rootPMD@root.domain.com

- Include only versions of the following policy:

  iis5

```
policyreport -name prodPolicies_March2006 -mode p \
-dms mainDMS@mainhost.domain.com -root rootPMD@root.domain.com -pn iis5#* \
-targetpath C:\eac_data\reports
```

The policyreport utility stores the report in a subdirectory we specified using the -name option (iis5Policies_March2006) under the directory we specified using the -targetpath option (C:\eac_data\reports). We can update this report at a later time by adding the -f option that creates the report even if one already exists in the output directory.

Example: Create a report for policies that were deployed with errors

The following example shows how you can use the policyreport utility to perform the following task:

- Generate a policy report in the following directory:

  C:\eac_data\reports\policyErrors

- Retrieve the information is from the DMS:

  mainDMS on the **mainhost.domain.com** computer.

- Include only computers (subscribers) that are hierarchically beneath the following PMDB:

  rootPMD@root.domain.com

- Include only policies that were deployed with errors (Failed)

```
policyreport -name policiesErrors -mode p \
-dms mainDMS@mainhost.domain.com -root rootPMD@root.domain.com \
-pstat "Failed" -targetpath C:\eac_data\reports
```

View a Policy or Host Report

Once you generate the report, you need to navigate to the folder where the report is stored and open it for viewing in a browser.

To view a policy or host report

1. Navigate to the folder where the report is stored.

   This is <*target_path*>/<*report_name*>/html

   where <*target_path*> is the directory you specified using the -targetpath flag and <*report_name*> is the report name you specified using the -name flag when generating the report.

2. Open the **index.html** file for viewing in your browser.

   The report's main page displays in your browser.

How Policy Deviation Calculations Work

Advanced policy management and reporting let you see the difference between the policy rules that are meant to be deployed on the end-point, and the actual policy rules that are applied on the end-point. This lets you deal with problems associated with the deployment of your policies.

The policy deviation calculation is run on each end-point and performs the following actions:

1. Retrieves from the local host the list of rules that should be deployed on the end-point.

   These are the rules that are specified for each of the deployed policies as specified in the local RULESET object that is associated with the POLICY object for each deployed policy.

2. Checks that each of these rules is applied to the end-point.

   Important! The deviation calculation does not check whether Windows (native) rules are applied. It also ignores rules that remove objects (user or object attributes, user or resource authorization, or actual users or resources) from the database. For example, the calculation cannot verify whether the following rule is applied:
   rr FILE C:\tmp\tmp.txt

3. (Optional). Compare between the policies associated with the local HNODE object and the one on the first available DMS.

   Normally, the deviation calculator checks for deviations only on the local host. If you specify the *-strict* option, this option, the deviation calculator also compares the local policies to the policies on the first available DMS in the list. It compares the:

   a. List of policies associated with the HNODE object representing the local host.

   b. Policy state of each POLICY object associated with the HNODE object.

   c. Policy signature of each POLICY object associated with the HNODE object.

4. Outputs the following two files:

   - *<eTrustACDir>*\data\devcalc\deviation.log

     Log and error messages collected during the last deviation calculation.

   - *<eTrustACDir>*\data\devcalc\deviation.dat

     List of policies and their deviations.

   Note: eTrust AC also sends audit events which can be viewed using **seaudit -a**. For more information about the seaudit utility, see the Reference *Guide*.

5. Notifies one or more DMSs of any deviations found.

The DMSs to notify are either specified manually (*-dms* option), or if no DMS is specified, the deviation calculator uses the DMS list specified for the local eTrust AC database.

Configure an End-point for Policy Deviation Calculations

When you install or upgrade eTrust AC using a Custom installation, the installation process configures the policy deviation calculator if you choose the Advanced Policy Management option. You can also configure the database to calculate and send deviation status notifications to specified DMS databases post installation.

To configure an end-point for policy deviation calculations

Note: You only need to do this if you did not install or upgrade this eTrust AC computer with the Advanced Policy Management option selected. For more information about eTrust AC installation, see the *Implementation Guide*.

1. Open a selang command window.

   A selang command window opens, letting you enter selang commands.

2. Enter the following command:

   ```
   nu ("+devcalc") admin auditor
   ```

   This creates a new user named *+devcalc* with the ADMIN attribute. This user is used by eTrust AC to run the deviation calculator.

3. Enter the following command:

   ```
   nr SPECIALPGM ("<devcalc.exe_path>") seosuid("+devcalc") nativeuid("SYSTEM")
   ```

   where *<devcalc.exe_path>* is the full path to the devcalc.exe application that resides in the bin directory of the eTrust AC installation directory.

   This creates a new special program resource for the deviation calculator and specifies the logical user authorized to run this special program and the native Windows user.

4. Enter the following command:

   ```
   so dms+(<DMS1>[,<DMS2>)
   ```

   where *<DMSx>* is the name of the DMS you want the deviation calculation to send policy deviation status notifications to. Each DMS has to be specified in the following format: DMS_name@hostname.

Example: Configure an end-point to send policy deviation status to a central DMS

The following example shows the commands you need to run on an end-point, which was installed using a Typical installation (and a default installation directory), to perform the following task:

- Configure the end-point to be able to perform deviation calculations.
- Send policy deviation status to the DMS:

  prodDMS on the **centralhost.com** computer.

```
nu ("+devcalc") admin auditor
nr SPECIALPGM ("C:\Program Files\eTrustAccessControl\bin\devcalc.exe") \
seosuid("+devcalc") nativeuid("SYSTEM")
so dms+(prodDMS@centralhost.com)
```

Policy Deviation Log and Error File

The policy deviation calculation writes a new log during each deviation calculation. This log also contains error messages and is stored in *<eTrustACDir>*\data\devcalc\deviation.log

Use this log when the deviations you see in your reports (that are retrieved from the DMS) are not gathered from the last time a deviation calculation should have run. It can help you diagnose why the deviation calculation results were not sent to the DMS.

Important! If the deviation log contains the error "ERROR: failed to initialize DB library, database is open", you need to recreate the database's index files. To do this, exit selang and run the following command from the *<eTrustACDir>*\data\devcalc\init_ac_db directory, then rerun the deviation calculation (see page 119):
selang -l -d .

Example: Deviation log and error file

The following is an example deviation log and error file:

```
start time: Mon Jan 23 13:04:48 2006
WARNING,\"failed to retrieve DMS host name, deviation will be stored locally\"
found deviation(s) for policy 'iis8#02'
end time: Mon Jan 23 13:05:04 2006
```

Policy Deviation Data File

The policy deviation calculation writes a data file that contains a list of policies and their deviations. This data file is stored in *<eTrustACDir>*\data\devcalc\deviation.dat

Note: The list of policies included in the data file depends on the policies that a deviation is calculated for (by default, all the policies and all policy versions on the end-point).

Important! The deviation calculation does not check whether Windows (native) rules are applied. It also ignores rules that remove objects (user or object attributes, user or resource authorization, or actual users or resources) from the database. For example, the calculation cannot verify whether the following rule is applied:
rr FILE C:\tmp\tmp.txt

The deviation status is sent (whether a deviation exists or not) to the DMS but the actual deviation is stored locally. When a report is created, the actual deviation results can be retrieved from this file and added to the report.

The following lines can appear in the policy deviation data file:

Date

Date of deviation calculation.

Format: DATE, *DDD MMM DD hh:mm:ss YYYY*

Strict

Specifies that the deviation calculation was run with the -strict option.

Format: STRICT, *DMS@hostname*, *policy_name#xx*, {1|0}

where {1|0} signifies whether a deviation was found (1) between the policies associated with the local HNODE object and the ones associated with the HNODE object on *DMS@hostname* (the first available DMS), or not (0).

Policy Start

Starts a policy block which defines the deviation for this policy.

Format: POLICYSTART, *policy_name#xx*

Difference

Describes a deviation that was found for a policy. The name of the policy for which the deviation applies to is the nearest **policy line** above this line.

There are four types of deviations which are described in the following table:

Deviation Type	Format
Class not found	DIFF, (<*class_name*>), (*), (*), (*)
Object not found	DIFF, (<*class_name*>), (<*object_name*>), (*), (*)
Property not found	DIFF, (<*class_name*>), (<*object_name*>), (<*property_name*>), (*)
Property value mismatch	DIFF, (<*class_name*>), (<*object_name*>), (<*property_name*>), *(<expected_value*>)

Policy End

Ends a policy block which defines the deviation for this policy.

Format: POLICYEND, *policy_name#xx*, {1|0}

where {1|0} signifies whether a deviation was found (1) or not (0).

Warning

Describes a warning.

Format: WARNING, "*warning_text*"

Example: Deviation data file

```
Date, Sun Mar 19 08:30:00 2006
WARNING, "failed to retrieve DMS host name, deviation will be stored locally"
POLICYSTART, iis8#02
DIFF, (USER), (am), (*), (*)
POLICYEND, iis8#02, 1
```

Perform a Deviation Calculation

A deviation calculation should be performed regularly so that the DMS contains recent information about policy deviation status. We recommend that you schedule a policy deviation calculation to occur in an interval that supports your reporting requirements.

To perform a deviation calculation on an end point, from a selang window, enter the following command:

```
start DEVCALC
```

Example: Schedule a routine deviation calculation

The following example shows how you can create on Solaris a deviation calculation task that:

- Runs daily at midnight.
- Sends the deviation status to the DMS:

  mainDMS on the **mainhost.domain.com** computer.

To do this, follow these steps:

1. Create a batch file with the following line in it:

   ```
   selang -c "start DEVCALC params('-dms mainDMS@mainhost.domain.com')"
   ```

2. Add a scheduled task making the following selections:
   - Browse and choose your new batch file.
   - Perform this task daily.
   - Start time: 12:00 AM

Integrate PMDBs with Unicenter

Integrating your PMDB with Unicenter TNG lets you use the PMDB to create rules that secure Unicenter TNG objects from being manipulated by the various Unicenter TNG components (such as the command processor, Event Management, and Workload Management).

You must perform the integration manually.

To integrate a PMDB with Unicenter TNG

1. Create the PMDB.
2. Migrate Unicenter Security options into the PMDB with the following command:

   `MigOpts` *pmdb-name*

   where *pmdb-name* is the name of your PMDB.

   Note: This step is required only if you used Unicenter Security and selected Security Data Migration under the Unicenter Integration during the eTrust AC installation. If you did not use Unicenter Security, then you never established any security options and there is nothing to migrate into your PMDB.
3. Create classes for any user-defined Unicenter TNG asset types with the following command:

   `defclass.bat.` *pmdb-name*

   where *pmdb-name* is the name of your PMDB

   Note: This step is required only if you used Unicenter Security and created user-defined asset types. Unicenter TNG asset types are automatically defined in every new PMDB if you selected Unicenter Integration during the eTrust AC installation.

Chapter 7: Using the Transaction Manager

This section contains the following topics:

The Transaction Manager

The Transaction Manager is a tool for managing eTrust AC, UNIX, and Windows security. It sends eTrust AC transactions to multiple hosts automatically, as they are performed on the local host. This transaction mode is intended as a quick and efficient substitute for, or adjunct to, the Policy Model. It does not offer the same guaranteed propagation of modifications to the security database of every subscriber, but it is simpler to use and is especially useful when you want to make changes to multiple databases that are not defined as part of your Policy Model hierarchy.

Setting Up the Transaction Manager

Before using the Transaction Manager, do the following:

1. Make sure you have ADMIN authority on each remote host you access, as well as on the local host.

   Create a TERMINAL record for the administering computer on each host you access.

   These requirements are the same as for administering a remote host.

   Enable the Transaction Manager. From Policy Manager, choose Tools, Options and open the Transaction Mgr tab. Check Enable multi-host transactions. You can also select the Transaction Manager options you want to activate.

2. Create the Target Host file.

Multi-Host Transaction Options

The Transaction Manager can be used with any window that can be opened from the program bar, with the exception of Security Policies and Audit. Users, Groups, and Resources are checked by default.

The Transaction Manager options let you customize the way transaction mode operates. The options are as follows:

General Options

Stop sending data at first error

The default is to continue sending transactions even if errors occur, which lets you propagate as many transactions as possible and deal with errors later. However, you can save time by terminating propagation at the first error when, for example, you are sending transactions to many hosts and you expect anything that fails on one to fail on the others.

Close Transaction Manager upon exit

Checking this box causes the Transaction Manager to exit along with Policy Manager, whether it has finished sending queued transactions. The default for this option is to keep Transaction Manager up even when Policy Manager is closed down.

Note: The Transaction Manager's memory is volatile; when it exits, its transactions log is lost.

Activate transaction mode upon startup

The Transaction Manager is always started when you start Policy Manager. However, by default, you are not in transaction mode until you click the transaction mode button on the toolbar. Checking this box puts you automatically into transaction mode when you start Policy Manager.

Target host file

This option lets you set the directory path for the target host file. The default path is *eTrustACDir*\data\hosts.txt (where *eTrustACDir* is the directory where you installed eTrust AC).

Interval (in seconds) between each refresh...

This option governs how closely the TM Status Window monitors the Transaction Manager. The default is 10 seconds, which may not show any progress if transactions are short.

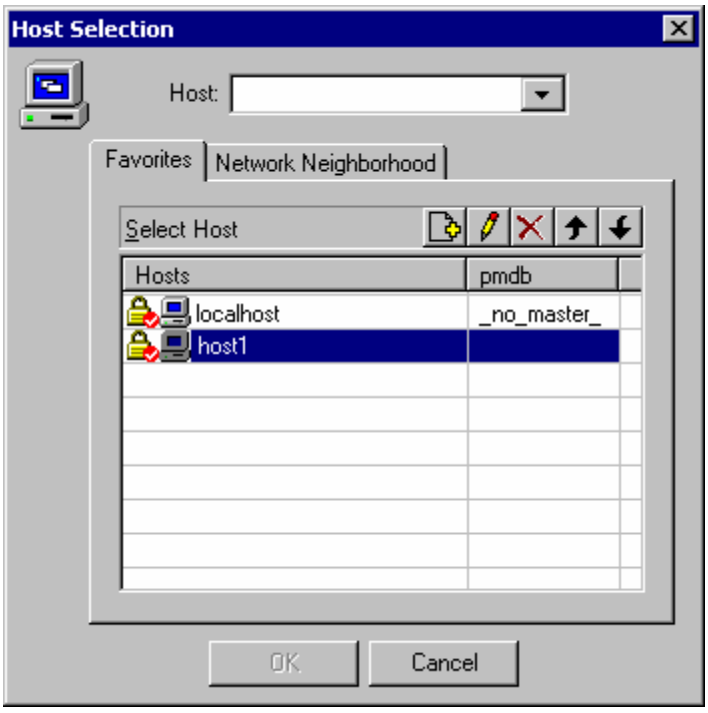
Commands and Scripts

Normally, you use transaction mode with Policy Manager transactions. You can also use it to send selang commands or scripts to multiple hosts. Checking these boxes allows transaction mode to be used with the Commands and Scripts dialog (Tools, Execute Command) to send selang commands to multiple hosts.

Setting Up the Target Host File

The Target Host file controls which hosts, or groups of hosts, receive transactions when you are working in transaction mode.

Add hosts from a Favorites list or from Network Neighborhood.



Hosts can be local databases or PMDBs. You can create groups of hosts to speed selection. Clicking a group name activates all members of the group. You can also select or deselect any individual host. The selections go into effect immediately after clicking OK and remain in effect until changed. You must manually reset the Target Host file each time you want to send transactions to a different group of hosts.

Note: When Transaction Mode is enabled, the Host Selection settings apply to the Copy User and Copy Group Wizards as well as the Transaction Manager.

Working in Transaction Mode

Once you have enabled the Transaction Manager and created the target host file, click the Transaction Mode icon on the toolbar. Any transaction you perform on the local database is propagated to the selected hosts as well. For example, when you delete a user by selecting a name in the Users window and clicking Delete on the toolbar, the transaction takes place immediately in the local host database (as usual) and is then automatically propagated to the hosts in the target hosts file.

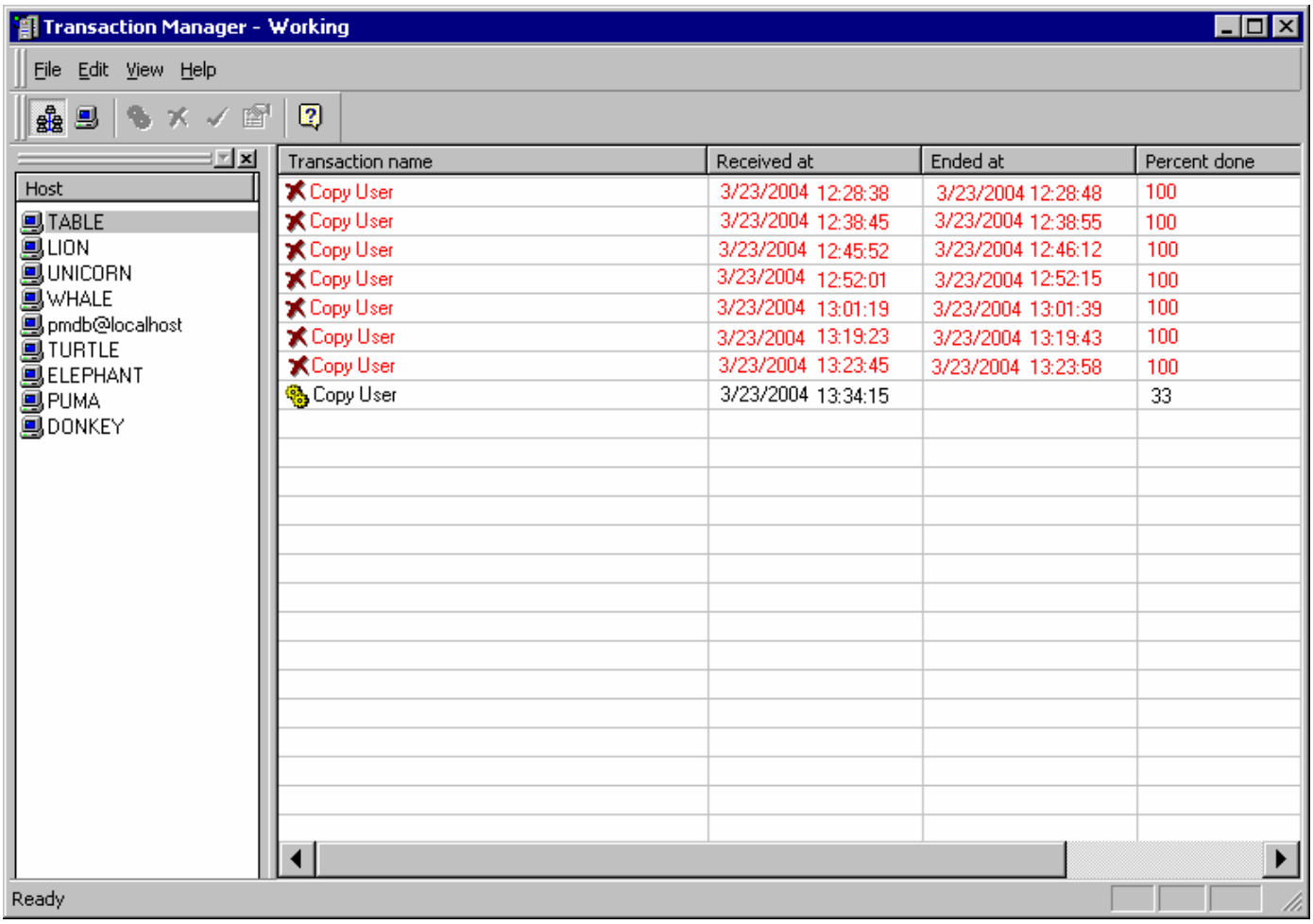
You can follow the progress of the propagation with the Task Manager Status window. To suspend lengthy transactions, right-click the Transaction Manager icon in the Taskbar tray and select Suspend Transaction Manager. Propagation is continued when you select Resume Transaction Manager. Closing the Transaction Manager window, even by clicking the close button in the upper right corner, does not terminate the application. To exit from Transaction Manager, right-click the icon in the Taskbar tray and choose Exit. Remember that, when you do this, you erase all transactions from the log.

The Transaction Manager Window

Double-click the Transaction Manager icon on the Windows Taskbar to activate the Transaction Manager window. This window contains a log of all transactions propagated to multiple hosts. The Task Manager's memory is volatile, so only transactions in the current session display. You can choose between two views, the Host Status and the Host Bar, from the View menu or by clicking a Host icon on the toolbar. Note that both of these icons are equivalent; clicking either one toggles the view. Buttons on the toolbar select the view, permit you to run a transaction again, delete or undelete it, or display its properties. When a queued transaction is deleted, Transaction Manager skips it and advances to the next transaction in the queue. At any time, you can undelete it to continue propagating the transaction. You cannot delete the current transaction.

Host Status Bar View

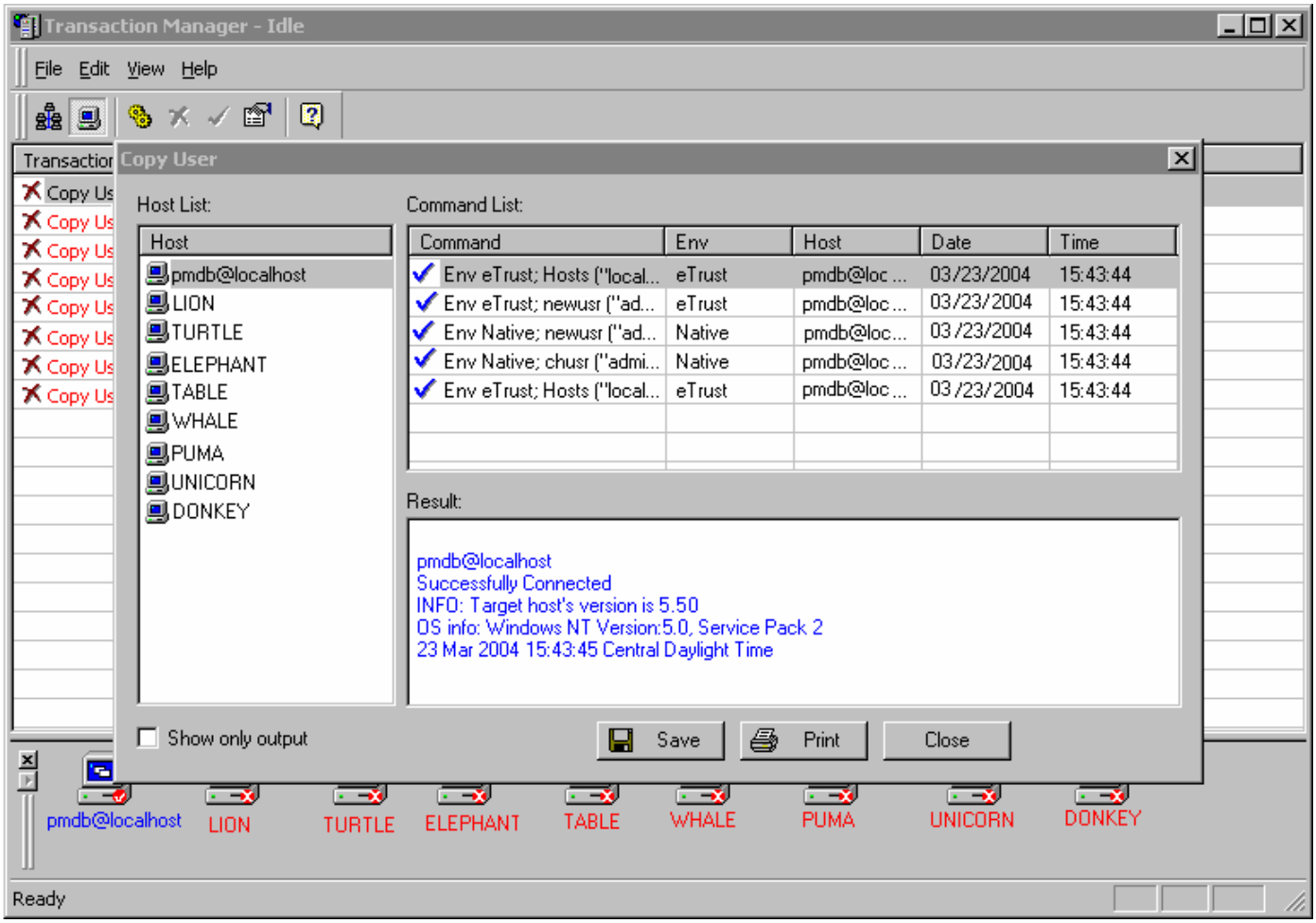
In the Host Status Bar view, an icon of each selected host appears in a status bar on the left side of the window. Clicking an icon relates the transaction list to that host. Double-clicking a transaction displays an Update dialog, with a Command list text box and a Result text box. Selecting a command in the Command list text box displays the results for that command in the Result text box. You have the option of printing the results or saving them to a file.



Host Bar View

In the Host Bar view, the transaction line runs the full width of the window and icons of the selected hosts display at the bottom. Double-clicking a transaction opens an update dialog containing a Host list bar, a Command list text box, and a Result text box. When you select a host icon, the Command list relates to that host. When you select a command, the Result relates to that command.

Alternately, you can select a transaction, and then double-click a host icon at the bottom of the window. The Update dialog appears.



Chapter 8: Monitoring and Auditing

This section contains the following topics:

Security Auditors

One of the most important tasks of security auditors and system administrators is auditing or monitoring system activity to detect suspicious or malicious activity. Security auditing plays an essential role in a secure environment, and the security auditing features in eTrust AC include the following:

- Providing a reliable indication of who has accessed the system, what resources have been accessed, and when resources have been accessed
- Notifying and alerting appropriate users in case of an attempted security breach, even if the attempt failed
- Indicating what changes have been made to the security rules, and by whom
- Providing a means to test the effect of access rules before they are enforced

eTrust AC auditing is modeled after real-world auditing: security auditors act independently of system and security administrators, although you can change your implementation so that this is not the case if some other model is more appropriate for your environment.

A security auditor is a user to whom the AUDITOR attribute is assigned. Users defined as security auditors are permitted to perform auditing tasks such as changing the audit rules that are assigned to users and resources. They are also authorized to use the eTrust AC auditing utilities without being required to have the ADMIN attribute.

Monitoring Access Control Activity

The eTrust AC trace is a realtime log that can show every action taken by eTrust AC. Trace records are accumulated in eTrustACDir\log\seosd.trace (where eTrustACDir is the directory where you installed eTrust AC).

Or they are accumulated in whatever file you specify as the *trace_file* value in the registry subkey:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrustAccessControl\SeOSD\
```

Although you can filter the records from the trace file, the trace mechanism is designed for system monitoring and not for security auditing.

By default, eTrust AC only generates trace messages during eTrust AC initialization. Once eTrust AC is initialized, it stops the trace mechanism and trace messages are not generated.

Filtering Trace Records

Using a trace filter file, you can specify that certain types of activity should not appear in the trace file. The trace filter file is specified with the *trace_filter* value in the registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrustAccessControl\SeOSD
```

The default value is *eTrustACDir*\log\trcfilter.ini (where *eTrustACDir* is the directory where you installed eTrust AC).

Important! eTrust AC creates the trace filter file at installation with the single line: *seosd.trace*. Never delete this record.

Each line in the trace filter file represents an access or an activity that should **not** be traced. For example, to eliminate tracing the users' access to Microsoft Word, add the following line to the trace filter file:

```
*winword.exe*
```

Filtering Audit Records

You can use the audit.cfg file (found in *eTrustACDir*\data) to filter audit records on a host by defining audit records that should not be generated. Each line represents a rule for filtering out audit information (that is, the audit records that match the criteria in the line will not appear in the audit file). This filter helps to limit the size of the seos.audit file by keeping only the records needed. You can set filtering rules for class name, object name, user name, program name, access rights, and authorization result. The eTrust AC engine (seosd) reads this file at startup.

Syntax

```
<class>;login info;<user>;<program-path>;<access-mode>;<auth-result>
```

Note: A ***** in any column is a wildcard meaning any value.

When a message is sent to the audit file, seosd checks whether the message matches one of the following entries in the audit.cfg file:

Field	Rule
Class	The class name should be written in upper case
Object	The name of the resource can be written using a pattern (*)
User	The user name can be written using a pattern (*)
Program path	The program being used can be written using a pattern (*)
Access mode	The access rights must adhere to the rules
Authorization result	The authorization result must be P (permit) or D (denied) **Note:** The value *P* also filters audit records that were generated for resources in warning mode.

Syntax for TCP class

```
<class>;login info;<host>;<program-path>;<access-mode>;<auth-result>
```

Example:Audit Access Filter

In the following example, if Administrator successfully reads a file, seosd does not send a message to the audit file. If Administrator cannot read the file, seosd sends a message to the audit file.

```
FILE;*;Administrator;*;R;P
```

Setting Audit Rules

For security auditing, eTrust AC keeps audit records for events of access denial or access grants according to the audit rules defined in the database.

Every accessor and resource has an AUDIT property that can be set to one or more of the following values:

FAIL

Logs access failures by the accessor to the resource.

SUCCESS

Logs successful accesses by the accessor to the resource.

LOGINFAIL

Logs every logon failure by the accessor. (This value does not apply to resources.)

LOGINSUCCESS

Logs every successful logon by the accessor. (This value does not apply to resources.)

ALL

Logs the same information as FAIL, SUCCESS, LOGINFAIL, and LOGINSUCCESS for accessors or FAIL and SUCCESS for resources.

NONE

Logs nothing concerning the accessor or resource.

Whenever you create or update an accessor or resource record in the database, you can specify the AUDIT property. You can also specify whether email notification of logged events should be sent and to whom. (You can use Policy Manager, as described in the chapter “Using the Administrator Interface” or selang commands, as described in the chapter “The selang Command Language” in the*Reference Guide,* to create and update records in the database.)

The records in the audit log accumulate according to these audit rules. The decision whether to log an event is based on the following:

- If the resource or accessor has AUDIT(ALL), all login events for the accessor and all events concerning resources protected by eTrust AC are logged, regardless of whether access failed or succeeded.
- If access to a resource protected by eTrust AC is successful and the accessor or resource has AUDIT(SUCCESS), the event is logged.
- If access to a resource protected by eTrust AC fails and the accessor or resource has AUDIT(FAIL), the event is logged.

Setting Audit Policies in Windows

In addition to setting access rules for accessors and resources, you can specify Windows events that you want to write to the audit log. You can specify such audit policies for the entire organization or on a user-by-user basis.

Audit Logs

The audit records created by events or accesses that you defined in the audit rules and audit policies constitute a file called the audit log. The value *audit_log* in the following Windows registry subkey specifies the location of the audit log:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrustAccessControl\logmgr
```

The default value for this key is:

```
*\Program Files\CA\eTrustAccessControl\log\seos.audit
```

By default, eTrust AC automatically backs up the audit log when it reaches 1024 KB by renaming the audit log file and creating a new one. You can change the audit-log size that triggers the backup by changing the value *audit_size* in the subkey:

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrustAccessControl\logmgr
```

You can also choose to back up the audit log periodically (daily, weekly, or monthly) by changing the value *BackUp_Date* in the Windows registry subkey:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\eTrustAccessControl\logmgr
```

Note: For more information about these registry subkeys, see the *Reference Guide*.

You should consider archiving old audit logs on tape to let you scan the events later.

Using Audit Logs

eTrust AC provides two built-in tools for viewing, filtering, and searching the audit logs:

- Policy Manager
- The seaudit utility

You can display every record in the audit log, or you can use filters to select particular records from the audit log.

The remainder of this chapter describes how to view the records in the audit log when using audit filters in Policy Manager.

Audit Filters

The number of records in the audit log can become enormous. To reduce the number of records that eTrust AC displays, use filters to select certain types of records for display. You can filter events by various criteria, including time or event type.

Create a filter in Policy Manager by assigning a name and choosing at least one switch. You can choose additional switches, and zero, one, or several options. You can also filter records with the seaudit utility.

Policy Manager provides several predefined filters, and you can create your own filters.

By default, eTrust AC stores all audit filters in *eTrustACDir*\data\AuditFilters.flt (where *eTrustACDir* is the directory where you installed eTrust AC).

You can choose to save the filters you create in this file, or you can save filters in another file.

Switches

INet host service

Lists the INET audit records of the TCP requests received from the specified hosts for the specified services. Host and service are masks that identify which set of hosts and services are searched for.

LOGON user terminal

Displays the following:

- LOGIN records for the specified user on the specified terminal. Both *user* and *terminal* are masks.
- Records created by the authorization engine when an invalid password is entered multiple times.

Resource class resource user

Lists resource records. You can specify the following:

- *Class*-a mask that identifies the class to which the accessed resource belongs.
- *Resource*-a mask that identifies the names of the resources that were accessed.
- *User*-a mask that identifies the name of the users who accessed the resources.

Start

Lists the startup and shutdown messages from the eTrust AC services.

Update (cmd, class, object, user)

Displays database update audit records. You can specify:

- *Cmd*-a mask identifying the set of selang commands to search for.
- *Class*-a mask identifying the classes to search for.
- *Object*-a mask identifying the records to search for.
- *User*-a mask identifying the users who executed the commands.

Watchdog

Lists the Watchdog audit records.

All

Lists all records except those sent to the audit log by the tracing facility.

Options

Ending date

Specifies the end date. Records logged after the specified date are not listed.

Ending time

Specifies the end time. Records logged after the specified time are not listed.

No failure

Specifies that failures not be listed.

No granted

Specifies that successful (granted) accesses not be listed.

No logout

Specifies that logout records not be listed.

Internet address

Specifies that Internet addresses be listed instead of host names in TCP/IP records.

No notify

Specifies that NOTIFY audit records not be listed.

No password

Specifies that password attempt records not be listed.

Origin *host*

Specifies that only records originating from the specified host be listed. This option is applicable only when connected to a UNIX workstation.

Starting date

Specifies the start date. Records logged before the specified date are not listed.

Starting time

Specifies the start time. Records logged before the specified time are not listed.

Show port number

Specifies that port numbers be listed instead of service names.

No warning

Specifies that warning records not be listed.

Predefined Filters

eTrust AC comes with the following predefined filters:

All

Displays every record in the audit log. No filtering takes place.

Today

Shows every record created today.

Records from the last 2 days

Shows every record created yesterday and today.

Records from the last 7 days

Shows the records created during the last seven days.

Connection to Access Control services

Shows records that indicate when users connect to eTrust AC services such as Policy Manager or selang.

Note: When connecting to a UNIX workstation, the name for this filter becomes Login Records. The records represent user logins.

Administration activity

Shows all records that update the eTrust AC or operating system databases. Updates to the databases include adds, deletes, and changes to all types of records.

UserDefined Filters

You can build as many filters as you need. You select the significant fields and name the filter. eTrust AC automatically saves the filter so that you can re-use it any time you invoke Policy Manager.

Warning Mode

As you phase in your security policy, you may find it useful to examine the behavior of certain resource access restrictions without actually enforcing the restriction. This practice is particularly useful:

- To determine if the rules you want to set are too strict or too lenient, so you can modify your security policy accordingly
- If you suspect that some restriction may have an adverse effect on the execution of a system application

eTrust AC lets you specify restrictions and substitute a warning message for the enforcement of the restrictions.

Implementing Warning Mode

To implement warning mode:

- Set the WARNING parameter in all the resource records affected by the rules you want to test.
- In Policy Manager, choose Auditing when you create or modify a resource and check the Warning box.
- In selang, use the warning parameter with the newres, editres, or chres command.

  Note: For more information see the chres/editres/newres commands in the *Reference Guide*.

When warning mode is turned on for a resource and the accessor is not authorized to access the resource in the requested manner, eTrust AC issues a warning message, logs the access (stating the violation was permitted because warning mode was in effect), and allows the access to the resource.

Notes:

- In Warning Mode, eTrust AC does not create warning messages for resource groups.
- If you use warning mode during eTrust AC implementation, make sure you have enough disk space for the audit logs, and make sure that the size limit of the audit log is large enough.

Chapter 9: Unicenter Migration and Integration

This section contains the following topics:

Installing Unicenter Integration Tools

eTrust AC is fully integrated into the Unicenter Enterprise Management environment. The following sections describe how eTrust AC handles the integration.

Important! To integrate Unicenter TNG with eTrust AC, you must have Unicenter TNG installed on the same machine as eTrust AC.

Note: For complete installation instructions for the Windows environment, see the *Implementation Guide*.

Unicenter Integration Features

The following sections describe how eTrust AC integrates with Unicenter TNG.

SSF/EMSec API Support

The EMSec APIs for Windows channel calls into a single DLL. The EMSec support for Unicenter Integration consists of a replacement CAUSECR.DLL. This DLL receives calls to the EMSec APIs, and then reformats and redirects these requests to equivalent eTrust AC APIs. The return code from the eTrust AC APIs are converted back into their corresponding EMSec API return codes and control is returned to the caller of the EMSec API. This approach protects the integrity of existing applications that are currently using the EMSec API.

EMSec support is active after the Unicenter integration setup procedure completes. The Unicenter integration setup replaces the current CAUSECR.DLL in the Unicenter installation path (CAIGLBL0000 directory). At this point, incoming EMSec API requests are intercepted by the replacement CAUSER.DLL and the requested information is seamlessly retrieved through the eTrust AC APIs.

Unicenter Security Data Migration Features

The following sections describe how to migrate Unicenter Security data to eTrust AC.

Unicenter Security Options Migration

The eTrust AC program called MigOpts.exe extracts selected Unicenter Security options and customizes the targeted eTrust AC database according to these options. To activate this feature, you must run the Unicenter integration with Unicenter Security data migration setup procedure. This setup procedure automatically runs MigOpts.exe.

Note: The following Unicenter Security options can be migrated **completely** into the eTrust AC environment:

- AUDIT_LOGIN
- MODIFY_PWDNEVEREXP
- PWDQUEUESIZE
- SEC_AUDIT_DBUPDATE
- SEC_AUDIT_SEND
- SEC_PASSWORD_ALPHA
- SSF_MAXPWDVIO
- SSF_MINPWDLEN
- SSF_SECPWEXCL
- USER_DEFSESID
- USER_PWDCHANGE
- USER_PWDCHGMAXDAYS
- USER_PWDCHGMINDAYS
- USER_PWDMAINT

Note: USER_PWDMAINT is migrated into the eTrust AC environment specific to the existing Unicenter Security data. eTrust AC maintains password information, but this process is not automated. When the existing Unicenter TNG users are exported from Unicenter Security into the eTrust AC database, this manual process is performed automatically if the value of the USER_PWDMAINT option is yes. However, after migration is complete, if an administrator adds a new user whose password information must be tracked, the administrator must additionally ensure that a "__workload__" application object exists. For example:

```
eTrust> na __workload__;
```

Then the administrator must update the user's login information to include the "__workload__" application object. For example:

```
eTrust> el (Username) appl('__workload__');
```

Additionally, ExportTngDb.exe migrates Unicenter TNG users who are members of the **SSF_AUTH** Unicenter Security option into the eTrust AC environment by setting the users' admin attribute before adding them to eTrust AC.

Unicenter Security Database Migration

The eTrust AC called ExportTngDb.exe extracts data from the Unicenter Security database and translates it into eTrust AC commands to populate the eTrust AC database. ExportTngDb.exe migrates the following:

- Unicenter Security users
- Unicenter Security user groups
- Unicenter Security rules

Notes:

- We do not recommend running Unicenter TNG login intercepts after running the Unicenter Integration and Migration Installation process. Once the Unicenter Integration and Migration Installation process has executed successfully, you should verify that Unicenter TNG login intercepts are disabled.
- Unicenter TNG Data Scoping rules (rules that target Unicenter TNG asset types with a -DT suffix) are not supported by the eTrust AC Migration process. Rules of this type are ignored during the migration process.
- Unicenter Security rules that have been implemented against any of the following Unicenter Security asset types are obsolete because Unicenter Security is no longer used: CA-USER, CA-ACCESS, CA-USERGROUP, CA-ASSETGROUP, CA-ASSETTYPE, and CA-UPSNODE. Rules that target any of these asset types, or any of their derivatives, are ignored during the migration process.

To activate ExportTngDb.exe, you must run the Unicenter integration with Unicenter Security data migration setup procedure. This setup procedure automatically performs the Unicenter Security data migration process.

Note: Creation and modification statistics of all Unicenter TNG objects are lost in the migration process.

Due to Unicenter TNG and eTrust AC product differences, the following attributes of Unicenter Security **users** cannot be migrated to eTrust AC:

Statistics

The following user statistics are not supported by eTrust AC:

- Last login statistics (date and time, node of last login)
- Password change statistics (date and time, node, user who changed last password, and expiration date of the Password)
- Password violation statistics (date and time, node of last unsuccessful login, and number of unsuccessful logins since last successful login)
- Access violation statistics (date and time, node of last access violation, and number of access violations)
- Suspension statistics (date and time of suspension)

PWDCHANGE VALUE (RANDOM)

Random password generation

UPSSTATGROUP

UPS station group

- Not supported by eTrust AC

USERORIGIN

User Origin (NIS or Local)

VIOLMODE

Violation Mode (FAIL, MONITOR, WARN, QUIET)

- eTrust AC supports FAIL mode only.

VIOLACTION

Violation Action (CANUSER, CANU&LOG, CANU&LOG&SUS)

- eTrust AC supports CANUSER action only.

Due to Unicenter TNG and eTrust AC product differences, the following attributes of Unicenter Security **rules** cannot be migrated to eTrust AC:

EXPIRES

Rule expiration date is not supported by eTrust AC.

Unicenter User Exit Support

To assist in migration, eTrust AC lets you run existing Unicenter Security user exits unchanged in the eTrust AC environment. You do not have to rewrite all user exits as part of the migration.

Using only the existing user exit interfaces in Unicenter Security and eTrust AC, each installed component is registered as a standard eTrust AC user exit, which then brings up the corresponding Unicenter Security exit.

To initiate this feature, run the Unicenter Integration with Unicenter Security Data Migration setup procedure. Once the setup procedure is complete, this functionality is active.

Note: Because Unicenter TNG and eTrust AC use different architectures, only the exit points and data items that are comparable between Unicenter Security and eTrust AC are supported. The following Unicenter Security exit points are supported:

EmSec_CredExit()

The input to the Unicenter Credential Authentication exit, EmSec_CredExit() is mapped by EMSECSIGNON. With eTrust AC, only the user and node members within this structure have meaningful data. The user member is set to the user name being authenticated, and the node member is set to the current local node name. All other members of the EMSECSIGNON structure are set to binary zeros. The other parameters, the detailed return code, and the message passed back from the Unicenter Resource Check Exit are ignored.

EmSec_PwExitNew()

The input to the Unicenter Password Validation exit, EmSec_PwExitNew, consists of a user (the user whose password we are changing), a password (the new password), and a node name (under e eTrust AC support, this is always the local node name). An older exit, EmSec_PwExit, is used if this fails. It contains only the user and password as input and is fully supported under eTrust AC.

EmSecSSFResCheck()

The input to the Resource Check exit, EmSecSSFResCheck(), is mapped by EMSECRESCHECK. The user member in EMSECRESCHECK is set to the value of the accessing user. The class member in EMSECRESCHECK is set to the value of the class of resource for the access. The entity member in EMSECRESCHECK is set to the value of the object name. The eTrust AC access information is converted to Unicenter-style access permissions and placed into the attributes member of EMSECRESCHECK. All other members of EMSECRESCHECK are set to binary zeros. The other parameters, the detailed return code, and the message passed back from the Unicenter Resource Check Exit are ignored. Once the Unicenter integration setup completes, this functionality is active.

Unicenter Calendar

Unicenter TNG provides a calendar facility, with which you can set time restrictions for users, groups, and resources. The calendar contains time intervals of 15 minutes that you can set to ON or OFF. A calendar time interval set to OFF prevents access to resource; a calendar time interval set to ON allows access to resource.

In Windows, an administrator can set calendar usage before security startup only.

Note: Unicenter TNG must be installed on the local machine. eTrust AC uses local Unicenter TNG services to retrieve calendar settings.

1. Stop eTrust AC security. Enter:

   ```
   secons -s
   ```

2. In the Windows registry, go to the following subkey:

   ```
   HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrustAccessControl\UCTNG
   ```

   In this subkey, set the TNG_calendars value to yes. Set the TNG_refresh_interval value to the appropriate time value in minutes.

3. Start eTrust AC security. Enter:

   ```
   seosd -start
   ```

To link an eTrust AC resource with the calendar, you must issue the following database commands from the prompt:

```
eTrust> nr CALENDAR calendar_name
eTrust> nr file C:\myfile.txt calendar (calendar_name) defaccess (a)
```

The Unicenter TNG calendar access control list (ACL) is an additional security constraint feature. The regular Unicenter TNG calendar property restricts the current resource according to the appropriate Unicenter TNG calendar status. The Unicenter TNG calendar ACL property restricts access for (or gives access to) specific users and groups for the current resource according to the Unicenter TNG calendar status.

The two types of ACL Unicenter TNG calendar properties are regular and restrictive.

- The regular calendar ACL property permits user or group access to the resource according to ACL access.
- The restrictive (denied) calendar ACL property denies user or group access to the resource accordingly to ACL access.

To add a user or group to the regular calendar ACL (CALACL), enter the following command in selang:

```
eTrust> auth resource_class_name object_name uid_or_gid_name calendar(calendar
name) access(access_value)
```

For example:

```
eTrust> auth file file1 uid(george) calendar(basecalendar) access(r w)
```

To add a user or group to the denied calendar ACL, enter the following command in selang:

```
eTrust> auth resource_class_name object_name uid_or_gid_name
calendar(TNG_calendar_name) deniedaccess(access_value)
```

For example:

```
eTrust> auth file file2 uid(george) calendar(holidays) access(r w)
```

You can use regular and restrictive properties for the same resource (such as calendar and uid). The following command adds a user named George with read access to the denied calendar ACL for file1.

```
eTrust> auth file file1 uid(george) calendar(holidays) deniedaccess(r)
```

To remove a user or group from a Unicenter TNG calendar ACL property, use auth- :

```
eTrust> auth- file file2 uid(george) calendar(holidays)
```

Use the Show Resource (sr) command to see all Unicenter TNG calendar ACLs assigned to a specific resource:

```
eTrust> sr file file1
```

Certification with Unicenter

- The following features comply with Unicenter TNG 2.2 SP1, Unicenter TNG 2.4, or Unicenter NSM 3.0:
  - Sending “events”
  - Synchronizing mainframe passwords
  - Using the Unicenter TNG calendar

Appendix A: Synchronizing Passwords with Mainframes

This section contains the following topics:

Password Synchronization Support

eTrust AC supports password synchronization between mainframes running eTrust CA-Top Secret Security, eTrust CA-ACF2 Security, or RACF security products and Windows or UNIX machines running eTrust AC. Synchronization is accomplished using the standard eTrust AC password Policy Model mechanism.

Password Policy Model Methods

To implement password synchronization with a mainframe in your network, choose a Windows machine running eTrust AC to serve as a parent to the mainframe and make sure the mainframe password synchronization option is installed. Next, define the mainframe to eTrust AC and subscribe the mainframe to the password Policy Model from that Windows machine. Once you have done this, any password change a mainframe user makes is propagated to all the machines in the password Policy Model hierarchy.

When you give the mainframe administrator eTrust AC authorization to make password changes, any user password change, suspend action, or resume user action the administrator takes on the mainframe is propagated from the mainframe through the password Policy Model hierarchy. Likewise, administrative password changes and suspend or resume user actions made anywhere in the password Policy Model hierarchy are propagated to the mainframe.

Installation Requirements for Password Synchronization

On the Mainframe

You must have Unicenter TNG 2.2 SP1, Unicenter TNG 2.4, Unicenter NSM 3.0, or CA Common Services installed on your computer. The password synchronization utility relies on Unicenter TNG's built-in CAICCI (Common Communication Interface).

You can find instructions for configuring the mainframe for password synchronization in the following locations:

- For eTrust CA-ACF2 Security, in the eTrust CA-ACF2 Security Administrator Guide
- For eTrust CA-Top Secret Security, in the *eTrust CA-Top Secret Security User Guide*
- For RACF, on the CA Common Services installation CD

On Windows

On each Windows machine that you want to use as a parent to a mainframe for password synchronization, you must install eTrust AC with the Mainframe Password Synchronization option.

Note: If you already installed eTrust AC, you can run the installation program again to choose the Mainframe Password Synchronization option. Reinstalling does not alter your current database or settings.

Before you begin the installation, obtain the host name, SYSID, and administrator name for each mainframe that you want to subscribe to the Policy Model from this machine. If you do not have access to this information at installation time, you can skip that part of the installation and subscribe the mainframes later.

To start the eTrust AC installation program, choose Custom Installation, and check the Mainframe Password Synchronization option. The installation wizard uses the Unicenter CAICCI package, but you must restart Unicenter TNG to update CAICCI configuration.

The installation lets you subscribe hosts to the Policy Model. If you have the host names and SYSIDs for the mainframes, you can subscribe them now. Otherwise, you can skip this step and subscribe these hosts later.

Checking the Installation

When you have completed the eTrust AC installation, you can check whether it successfully installed the necessary services and processes as follows:

1. Use the Windows Services applet (Start, Settings, Control Panel, Services for Windows NT or Start, Settings, Control Panel, Administrative Tools, Services in Windows 2000) to view the list of services.

   The following services should appear in the Services list:

   - Unicenter (NR-Server)
   - Unicenter (Remote)
   - Unicenter (Transport)

   eTrust AC Main Frame Sync

2. Open the Windows Task Manager and choose the Processes tab.

   The following processes should appear in the list:

   - mfscpfd.exe
   - mfsd.exe
   - eacmfs.exe

If you subscribed mainframe hosts to a Policy Model during installation, verify that they appear in the subscriber list as follows:

1. From the Start menu, choose Programs, CA, eTrust Access Control, Policy Manager.
2. Click the Tools button at the bottom of the left panel.
3. Click the Policy Model icon.
4. In the tree view, choose the Policy Model to which you subscribed mainframes during installation.
5. Verify that the mainframe hosts you subscribed appear in the list on the right.

Completing the Policy Model Configuration

Once you have installed the appropriate software on your mainframes and the parent Windows system, you must perform the following actions on the Windows system to complete the configuration required for password synchronization:

- Create an MFTERMINAL record in the PMDB for each mainframe host that you plan to subscribe to the Policy Model.

  When you create a record in the PMDB instead of the local database, this record gets propagated to all the hosts in the Policy Model hierarchy.

- Create a USER record in the PMDB and in the native Windows environment for each mainframe administrator who issues password changes, giving these users Administrator or Password Manager authority so that eTrust AC recognizes their right to make password changes.

- Again in the PMDB, give these mainframe administrators full access permissions (read and write) to the TERMINAL record for the Windows machine and read access to the MFTERMINAL records for any mainframe from which they can issue a password change.

- Give these mainframe administrators logon locally privileges in the native Windows environment. Password changes that arrive from the mainframe must be able to be executed in eTrust AC on the local machine under the authorization of the appropriate mainframe administrator user.

- Subscribe the mainframes to the Policy Model (if you did not do this during installation).

- Check the passwd_pmd key in the Windows registry to ensure that it specifies the password Policy Model for password changes from the local machine and its subscribers. Update the registry entry if necessary.

These items do not need to be performed in any particular order (except, of course, that you cannot give mainframe administrators access permissions to the MFTERMINAL records until you have created the administrator's user record and the mainframe's MFTERMINAL record).

The following procedure is one way to accomplish these steps.

1. From the Start menu, select Programs, CA, eTrust Access Control, Policy Manager.
2. If you did not subscribe the mainframes to a Policy Model during installation, follow these steps. Otherwise, skip to Step 3.
   a. Click the Tools button in the program bar on the left.
   b. Click the Policy Model icon.
   c. Select the appropriate Policy Model.
   d. From the Edit menu, choose Add Subscriber.

e. For Subscriber Name, enter the fully qualified host name for the mainframe.

f. Check the Mainframe Subscriber box.

g. Select the host type for this mainframe (ACF, ACF2, RACF, TNG, or TSS) and enter the mainframe's SYSID and administrator name.

h. Click OK.

i. Repeat these steps for each mainframe subscriber you want to add.

3. Check the passwd_pmd key as follows:

a. Click the Windows NT button in the program bar on the left.

b. Click the registry Editor icon.

c. From the tree view, navigate to the following:

`HKEY_LOCAL_MACHINE\Software\ComputerAssociates\eTrustAccessControl\eTrustAccessControl`

d. Double-click the passwd_pmd key in the right-hand list.

e. In the Value area, select the String button and type the fully qualified name of the local machine's parent in the password Policy Model hierarchy, if these items are not already correctly specified.

f. Click OK.

4. Create USER records for the mainframe administrators, give them the authority to change passwords in eTrust AC, and give them the logon-locally user right in Windows, as follows:

a. Choose File, Connect, or click the Connect button on the toolbar.

b. In the Host Selection dialog, select localhost pmdb or enter *pmdbName*@localhost in the Host text box, where *pmdbName* is the appropriate Policy Model.

c. Click OK.

d. Click the Access Control button in the program bar on the left.

e. Click the Users icon.

f. Click the New button on the toolbar.

g. In the Create New User--General dialog, enter the name of the mainframe administrator.

h. Click the User Attributes icon.

i. In the User Attributes dialog, check Administrator or Password Manager, depending on the authority you want to give this user.

j. Click the Miscellaneous icon.

k. In the Miscellaneous dialog, click the User Privileges button.

l. From the list of available user rights, select Logon locally and click the >> button to move this user right to the Privileges Granted list.

m. Click OK to close the User Privileges dialog.

n. Click OK to add this user.

o. Repeat steps e through n for each mainframe administrator.

p. Click the Resources icon.

q. From the tree view, choose Logon Protection, Terminal.

r. From the list of TERMINAL records, choose the record for the local machine.

s. In the View or set TERMINAL properties dialog, click the Authorize icon on the left.

t. Click the New button to the right of Add Accessors.

u. Browse the list of accessors and select the mainframe administrator from the list.

v. Click OK.

w. In the Permissions area, select the All button.

x. Repeat steps e through i for each mainframe administrator.

y. Click OK.

5. Close Policy Manager and start selang.

6. Connect to the Policy Model:

```
eTrust> host pmd@localhost
```

7. Create an MFTERMINAL record for each mainframe using the following command:

```
eTrust> newres MFTERMINAL mfSYSID defaccess (none) owner (userName)
```

where *mfSYSID* is the SYSID of the mainframe and *userName* is the user who owns this MFTERMINAL record.

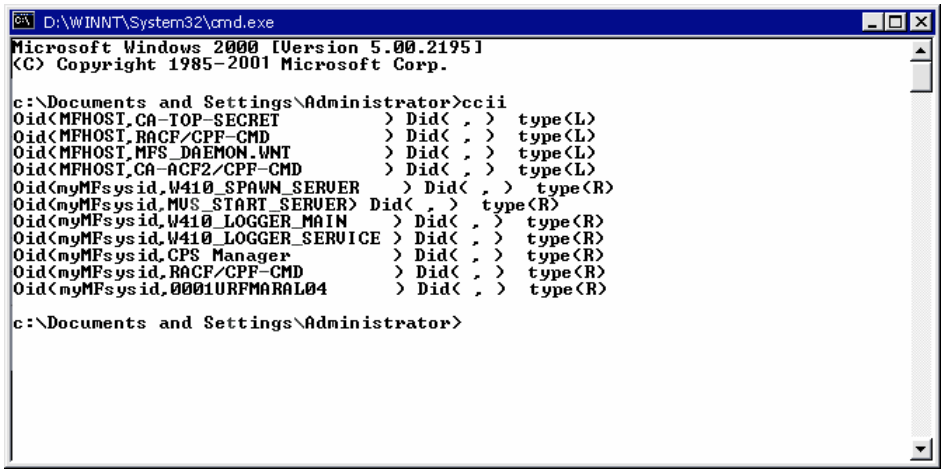
8. Give each mainframe administrator access to the appropriate MFTERMINAL record using the following command:

```
eTrust> authorize MFTERMINAL mfSYSID uid (mfAdmin) access (read)
```

where *mfSYSID* is the SYSID of the mainframe and *mfAdmin* is the mainframe administrator user.

Starting Mainframe Synchronization

To be sure that communication was established, run the ccii utility from a command prompt.



The CAICCI Configuration File

During installation and anytime you subscribe a mainframe to a Policy Model, eTrust AC automatically updates the CAICCI configuration file.

If, for some reason, you want to perform these updates manually, use the following procedure:

1. In Notepad, open the CAICCI configuration file (*cciDirectory*\tng\caiuser\ccirmtd.rc).
2. Add the following line:

   `REMOTE = mfName mfSYSID 1024 startup port 1721`

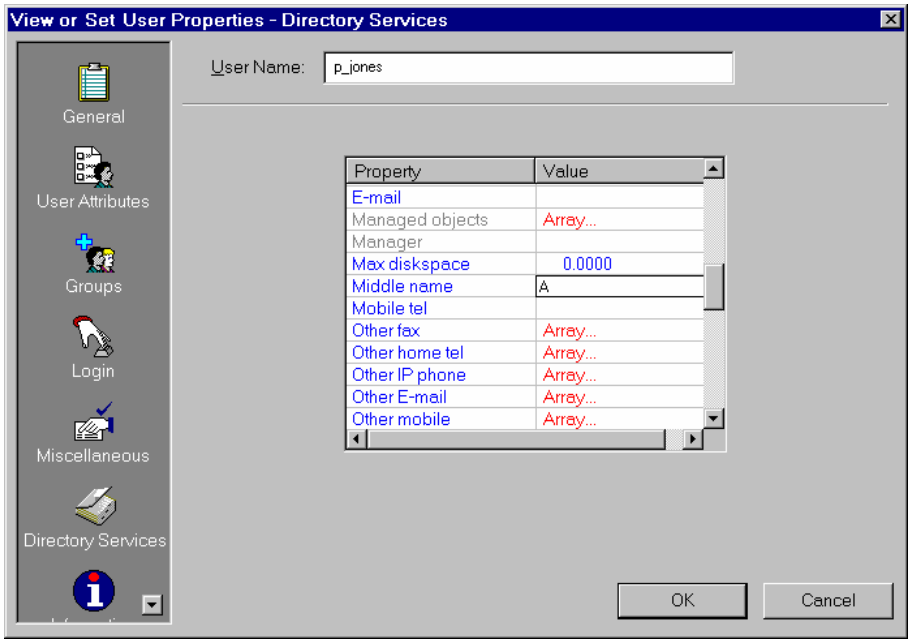
   where *mfName* is the mainframe name and *mfSYSID* is the mainframe SYSID.
3. Save the file.
4. Stop remote CAICCI services:

   `ccicntrl stop rmt`
5. Restart remote CAICCI services:

   `ccicntrl start rmt`

Set Active Directory User or Group Properties

When you are connected to a Windows 2000 machine with Active Directory, you can use the Directory Services panel of the User or Group Properties dialog to set Active Directory User or Group properties. These properties are not supported in native Windows NT, in Windows 2000 without Active Directory, or in the eTrust AC database environment.



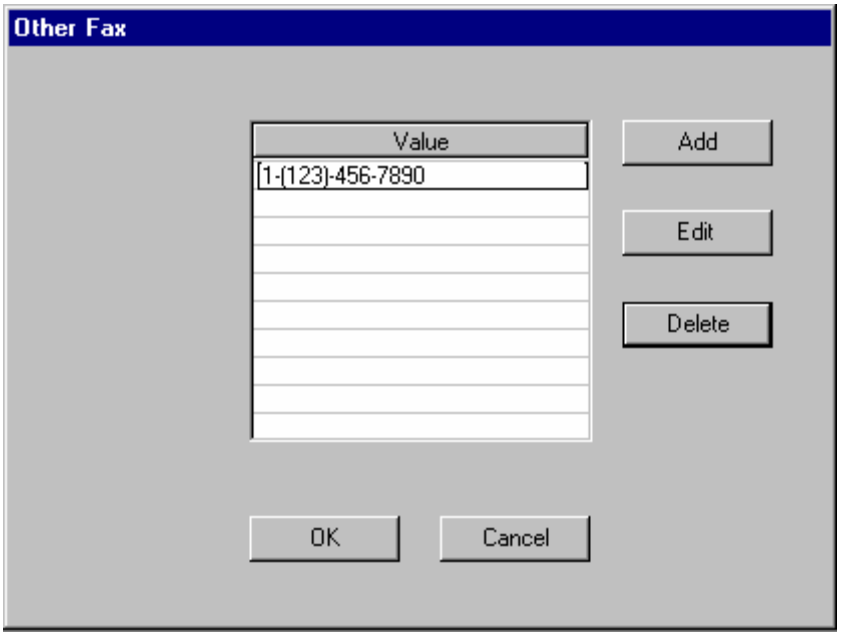
If you do not have Active Directory in your computer, connect to one that does. (It must have eTrust AC installed.)

1. Create a new user, and open the Directory Services panel.

   Note: The icon to activate the panel does not appear unless you are connected to a Windows 2000 machine with Active Directory.

2. Scroll down through the list. When you want to make an entry, double-click on the Value (right) side of the table. A box appears around the cell, and you can type into it.

   Note: If you double-click where the word Array appears, a dialog displays with a table to fill in.



Index

G

H

I

K

L

M

N

O

P

R

S

T

U

W