



The Compliant Enterprise Active Control System

Release 2.0

Implementation Guide



May, 2007

Copyright © 2006-2007 NextLabs, Inc. All rights reserved.
The information in this document is subject to change without notice.

NextLabs welcomes comments or suggestions regarding this manual or any of our product documentation. Please send an e-mail to techpubs@NextLabs.com.

TRADEMARKS

Compliant Enterprise™, ACPL™ and the Compliant Enterprise logo are registered trademarks of NextLabs, Inc. All other brands or product names used herein are trademarks or registered trademarks of their respective owners.

LICENSE AGREEMENT

This documentation and the software described in this document are furnished under a license agreement or nondisclosure agreement. The documentation and software may be used or copied only in accordance with the terms of those agreements. No part of this manual may be reproduced, stored in a retrieval system or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's use, without the prior written permission of NextLabs, Inc.

Published in San Mateo, CA, by NextLabs, Inc.
www.nextlabs.com
info@nextlabs.com
408.919.0191

Document Revision Number: ImpG2.0-B02

| | |
|---|-----------|
| Preface | 7 |
| Who Needs This Manual? | 7 |
| Product Documentation | 8 |
| Product Overview | 8 |
| Getting Started Guide | 8 |
| Implementation Guide | 9 |
| System Administrator's Guide | 9 |
| Policy Author User's Guide | 9 |
| Enforcer Administrator's Guide | 9 |
| CE Reporter User's Guide | 10 |
| Solutions Guide | 10 |
| Current Versions | 10 |
| Release Notes | 10 |
| Feedback | 10 |
| 1. Hardware Sizing | 11 |
| Hardware Requirements | 11 |
| 2. The Implementation Process | 13 |
| Overview | 13 |
| Information Control Solutions | 13 |
| The Process | 15 |
| 3. Information Activity Auditing | 17 |
| About Auditing | 17 |
| The Spectrum of Users | 17 |
| Auditing and the Active Control Solutions | 18 |
| Auditing Strategies | 18 |
| Stage and Scope | 19 |

| | |
|--|-----------|
| Strategy A: Global Auditing | 20 |
| Running a Global Audit | 20 |
| Global Audit Reports | 20 |
| Limitations | 21 |
| Strategy B: Targeted Auditing | 22 |
| What to Target | 22 |
| How to Target | 23 |
| Limitations | 23 |
| Using Silent Mode | 24 |
| Strategy C: Focused Auditing | 25 |
| Running Focused Audits | 25 |
| Focused Audit Reports | 26 |
| Strategy D: Security Auditing | 27 |
| Summary | 28 |
| Components, Policies, and Audits | 29 |
| Ongoing Audits | 30 |
| 4. Policy Planning | 31 |
| The Policy Planning Process | 31 |
| Virtual Information Barriers | 33 |
| Enter Compliant Enterprise | 34 |
| Designing Barriers | 35 |
| Understanding Information Control Requirements | 36 |
| Clarify the Problem | 36 |
| Assigning Control Categories | 37 |
| Color Codes | 38 |
| About Release | 39 |
| The Time Factor | 39 |
| Defining Control Levels | 40 |
| Refining the Rules | 40 |
| Virtual Barriers | 41 |
| About Costs | 42 |
| Measuring Up | 42 |
| About Network Audits | 43 |
| Moving On | 43 |
| 5. Policy Set Design | 45 |
| Components and Policies | 45 |
| Typical Policy Goals | 46 |
| Policy Design | 46 |
| Design Sequence | 48 |
| What You Can Do | 49 |
| Components | 49 |

| | |
|--|-----------|
| Action Components | 49 |
| Object Components | 49 |
| Policies | 50 |
| Choosing a Policy Effect | 51 |
| Operators | 51 |
| Nested Properties | 52 |
| Use Wildcard Characters | 52 |
| Control Use Schedules | 52 |
| Create Lock Boxes | 53 |
| General Considerations | 53 |
| Custom Obligations | 53 |
| Using Silent Mode | 54 |
| “Directory Components” | 55 |
| Using Active Directory Synchronization | 56 |
| About Policy Sets | 58 |
| Defining a Base Policy | 58 |
| Designing Fill-in Policies | 64 |
| Allow Only vs. Deny Policies | 64 |
| Design Questions | 64 |
| Evaluating your Fill-in Policies | 66 |
| Defining Lock Policies | 66 |
| About Directory Locking | 67 |
| Looking for Conflicts | 67 |
| Guidelines for Design | 68 |
| Designing Policies | 68 |
| Designing Components | 70 |
| Policy Set Examples | 71 |
| 1. Access & Use Control | 71 |
| 2. Duplication Control | 72 |
| 3. Export Control | 72 |
| 6. Deploying Policies | 73 |
| The Staging Environment | 73 |
| Limited Deployments | 73 |
| Staging Systems | 73 |
| Refining and Redeploying | 74 |
| Testing Policies | 75 |
| Managing Policy Versions | 76 |
| Versions and Audits | 76 |
| 7. Monitoring Your System | 77 |
| Monitoring Compliant Enterprise | 77 |
| Monitoring Policy Enforcement | 79 |
| About Reports | 79 |

- Report Types80
 - Action Filters80
- Using Reports82
- Sample Reports83
- The Standard Debugging Procedure 85
 - Where to Start85
 - Identifying Technical Problems86
 - Identifying Design Flaws95
- Index97**

Preface

Welcome to the Compliant Enterprise Active Control System, the information control platform that provides broad insight into and control over how information is used in your enterprise. Compliant Enterprise not only protects information from unauthorized access, it lets you control how files are used after access is granted.

Who Needs This Manual?

When Compliant Enterprise is installed in an organization, someone will be responsible for defining and clarifying information control policies, and someone will need to implement these policies in the Compliant Enterprise policy definition tool, Policy Author. The Compliant Enterprise documentation tries to emphasize this distinction, as one between design and implementation. The former does not necessarily require any proficiency with Policy Author, but rather requires a detailed knowledge of the organization, its network resources, and its information control requirements. The actual construction of policies and components is carried out in Policy Author, and focuses very much on that application; complete information on its features and how to use them are provided in the *Policy Author 2.0 User's Guide*.

Designing policies does not require technical knowledge of Compliant Enterprise per se, but it does require an understanding of compliance and information control issues and the business processes of your organization. Therefore, the task of planning and defining policies is likely to be assigned to someone with experience in business analysis, familiarity with security concepts, and a widespread knowledge of the company's business processes. The construction and deployment of policies in Policy Author could be carried out by someone else. For example, your legal department or compliance team could create the policies, then give the policy descriptions to technical specialists who will then use Policy Author to actually construct the policies.

Typically, a company's IT department will continue to be responsible for managing devices used by the organization, and will already be thoroughly familiar with the topography of the organization's resources. This knowledge is indispensable for properly defining policy components and manage policy deployment. Therefore, the tasks of modeling and deployment might naturally fall to personnel from the IT department.

The division of tasks among Policy Author users depends on the needs of the organization. In one organization, the design, policy construction, and deploy-

ment tasks might each be carried out by different people, while in another organization they all might be done by one person. The division of labor can be accomplished by customizing Compliant Enterprise user roles. Compliant Enterprise includes two pre-defined roles that are meant for Policy Author users: Policy Analyst and Policy Administrator. Typically, users with the Policy Analyst role are responsible for writing policies, while those with the Policy Administrator role are responsible for deploying policies and modeling policy components.

Product Documentation

The Compliant Enterprise documentation set consists of eight titles: an introductory *Product Overview*; a *Getting Started Guide* with installation and configuration instructions; an *Implementation Guide* to help with strategies for auditing information use and designing policies; an administrator's guide for all enforcers and one for the system overall; user's guides for Policy Author and Reporter; and a guide to the predefined active control solutions available with release 2.0.

Product Overview

Because Compliant Enterprise is a powerful, distributed enterprise product, its components are likely to be used by a number of different users in any given organization. Even though various users may be engaged exclusively with individual components of the suite and may not be interested in any others, we strongly recommend that all users read the Product Overview carefully, in order to acquaint themselves with the high-level architecture and function of the platform as a whole.



Getting Started Guide

The *Getting Started Guide* provides instructions on planning your system architecture and installing the Control Center and Policy Author.

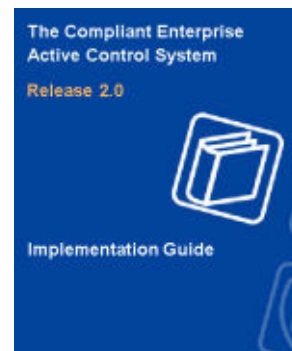
The installation procedures for all policy enforcers are provided separately, in the *Enforcer Administrator's Guide*.

Instructions on enrolling network entities, which is required after installation, are also provided separately, in the *System Administrator's Guide*.

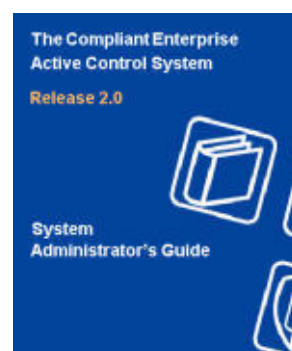


**Implementation
Guide**

This *Implementation Guide* provides a high-level approach to designing and implementing the information control policies that best suit your enterprise's needs. It offers generic advice on analyzing your needs through information use audits, approaches to designing appropriate policies, and optimizing those policies based on ongoing monitoring.

**System
Administrator's
Guide**

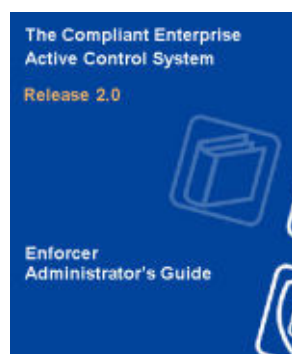
The *System Administrator's Guide* provides information required for managing and maintaining the Compliant Enterprise system once it is set up. It provides complete instructions on enrolling all kinds of network entities, which is required after the initial software installation. It also includes all user information for the administrative web application called Administrator, as well as for all utilities and other tools provided with the product. It is directed at the IT specialists who will be responsible for maintaining the Control Center after it has been installed.

**Policy Author
User's Guide**

The *Policy Author User's Guide* provides complete information on how to use Policy Author, the user interface where you build, deploy, and manage your information control policies and the library of policy components they are built upon. It is intended for the Compliant Enterprise user who will be responsible for converting generically expressed information policy goals into the specific, ACPL-based policy controls that are actually distributed to enforcement points throughout the enterprise.

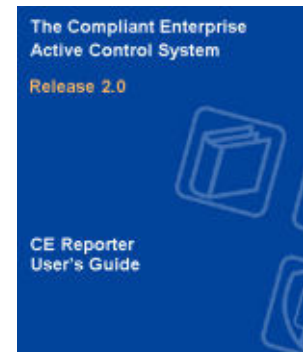
**Enforcer
Administrator's
Guide**

The *Enforcer Administrator's Guide* provides information on installing, using and maintaining all the types of enforcers currently available for Compliant Enterprise: for Windows file servers, Linux file servers, Windows desktops, and SharePoint servers. It is intended for the technical specialists who will be managing the enforcers; these may be the same as the Control Center administrators, or they may be different.



CE Reporter User's Guide

The Reporter User's Guide provides complete information on how to use Reporter, the web-based application that lets you easily generate reports on information use and access in your enterprise, and on the performance of your deployed policies. It is required reading for anyone with permission to generate or view Compliant Enterprise reports.



Solutions Guide

The Solutions Guide provides detailed information on customizing and using the pre-designed Active Control Solutions that are included with Compliant Enterprise: for Information Entitlements, for Endpoint Data Protection, and for Business Information Barriers.



Current Versions

Documents distributed in PDF format can become obsolete as subsequent versions are released. If you would like to check whether you are using the most current version of this or any manual, check the Document Control Number (DCN) at the bottom right of the inside cover, then click [here](#) to view a table of the most current versions of all Compliant Enterprise manuals. If the version listed in that table is later than the one in this manual, contact info@nextlabs.com to request the more recent version.

Release Notes

The release notes for each release of Compliant Enterprise are available directly on the installation CD, from the link on the splash screen or from the Docs directory. They describe any features or changes that could not be included in the documentation, and provide a list of known problems with the current version, along with suggested workarounds when appropriate.

Feedback

Feedback from Compliant Enterprise users is a valuable resource in helping our Product Information group provide you with the highest quality documentation as our product line develops. To this end, we would appreciate any comments you have on this manual or on any other Compliant Enterprise documentation; please send all feedback to info@nextlabs.com.

Hardware Sizing

This chapter provides help on estimating the hardware you will require for your implementation of Compliant Enterprise.

Hardware Requirements

Figure 1-1 represents the standard architecture of a standard installation of Compliant Enterprise. The hardware requirements for each component are described [Table 1-1](#).

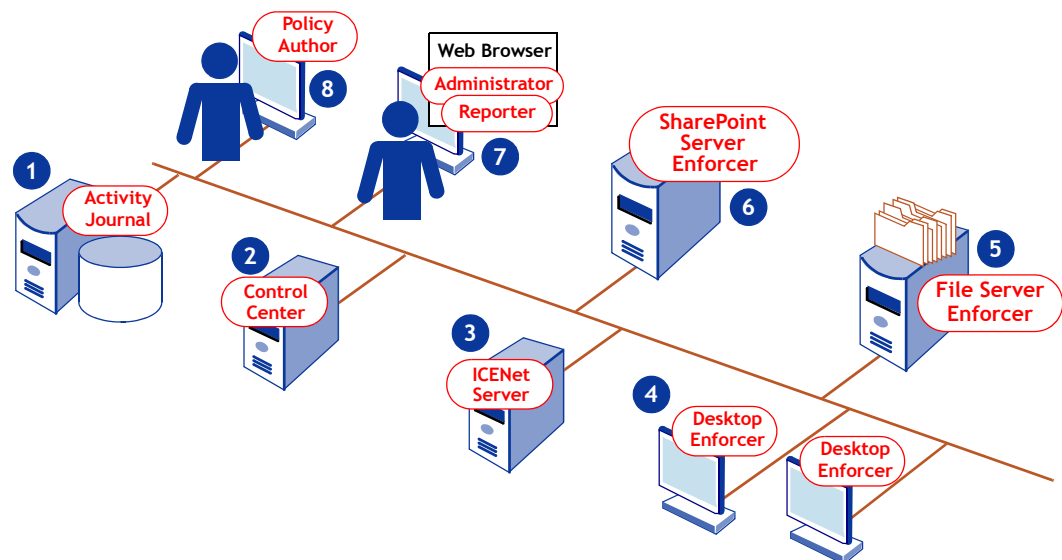


Figure 1-1: Compliant Enterprise Implementation: Hardware Requirements

Table 1-1: Compliant Enterprise System Requirements

| Component | OS | Processor | RAM | Disk Space | Comments |
|--------------------------------|---|-----------------|--------|---|---|
| (1) External Database | n/a | n/a | n/a | | Supported Databases: <ul style="list-style-type: none"> • Oracle 10g • PostgreSQL 8.0 |
| (2) Control Center | Windows Server 2003 | Pentium 1.8 GHz | 1 GB | 500 MB for server components, plus additional space for data logs | This is the recommendation for each host machine, regardless of the distribution of Control Center server components. The ICENet server component should be installed on a separate host. |
| (3) ICENet Server | Windows Server 2003 | Pentium 1.8 GHz | 2 GB | | The ICENet server is the component most sensitive to system load. Maximum capacity per server is between 1500 and 3000 desktop enforcers; this range depends on the type and volume of policy enforcement activity. For additional enforcers beyond this maximum, you should install additional ICENet servers using a standard load sharing appliance. |
| (4) Desktop Enforcer | Windows 2000, Service Pack 4 or higher, or Windows XP, Service Pack 1 or higher | Pentium 600 MHz | 256 MB | 120 MB | |
| (5) File Server Enforcer | Windows Server 2003 | Pentium 600 MHz | 256 MB | 120 MB | For calculating system load, consider each File Server Enforcer as roughly equivalent to two Desktop Enforcers. |
| | Red Hat Linux Enterprise Server Release 1, Updates 1-3 | | | | |
| (6) SharePoint Server Enforcer | n/a | n/a | n/a | n/a | Requirements for enforcer are same as for running the SharePoint Server itself. |
| (7) Policy Author | Windows 2000, Service Pack 4 or higher, or Windows XP, Service Pack 1 or higher | Pentium 600 MHz | 256 MB | 120 MB | |
| (8) Reporter and Administrator | n/a | n/a | n/a | n/a | Supported Browsers: <ul style="list-style-type: none"> • Microsoft Internet Explorer 6.0 or higher • Mozilla 1.7-based browsers, including Firefox and Netscape Navigator |

The Implementation Process

This chapter provides a brief overview of the process of the planning, designing, and rolling out your Compliant Enterprise implementation. More detail about each of these steps is provided in later chapters in this manual. For detailed discussions of all mechanical issues such as installation, configuration, and using Policy Author, Administrator and Reporter, refer to your *Compliant Enterprise System Manual*.

- Overview
- The Process ([page 15](#))

Overview

All work you do in Compliant Enterprise involves achieving some objective. Depending on what that is, in the course of solving it you may need to define one policy or a set of policies, or you may not need any policies.

In any case, you start with a simple question: what is my objective here? Every user will answer this in one of two basic ways:

1. **Audit:** I need to analyze my organization, looking for any problems.
2. **Control:** I need to address a problem I already know about.

These basic objectives represent the two strategic branches of the process represented in [Figure 4-1](#). The auditing branch consists of identifying which of four basic strategies best matches the kind of auditing you need to perform. This process will be covered in Chapter 3. The control branch consists of a sequential process of clarifying each individual problem separately and, based on the concept of a *virtual information barrier*, homing in on a set of policies and components that will solve it. This process will be covered in Chapter 4 and Chapter 5.

Information Control Solutions

If you already have a reasonably clear idea of the problems you want to solve, you may benefit from using one of the pre-designed information control solutions included with Compliant Enterprise. Each solution offers a set of already defined components, policies, and reports specifically designed to solve a cer-

tain information control problem. To consider whether these solutions would be helpful in your case, refer to the *Compliant Enterprise 2.0 Solutions Guide*.

Note, however, that even if you plan to use one of the pre-designed solutions, you will have to spend some time customizing and fine-tuning the solution policy sets. For this reason, every kind of policy planner will benefit from reading this manual, in order to gain a broad understanding of how to design, use, test, and refine policies.

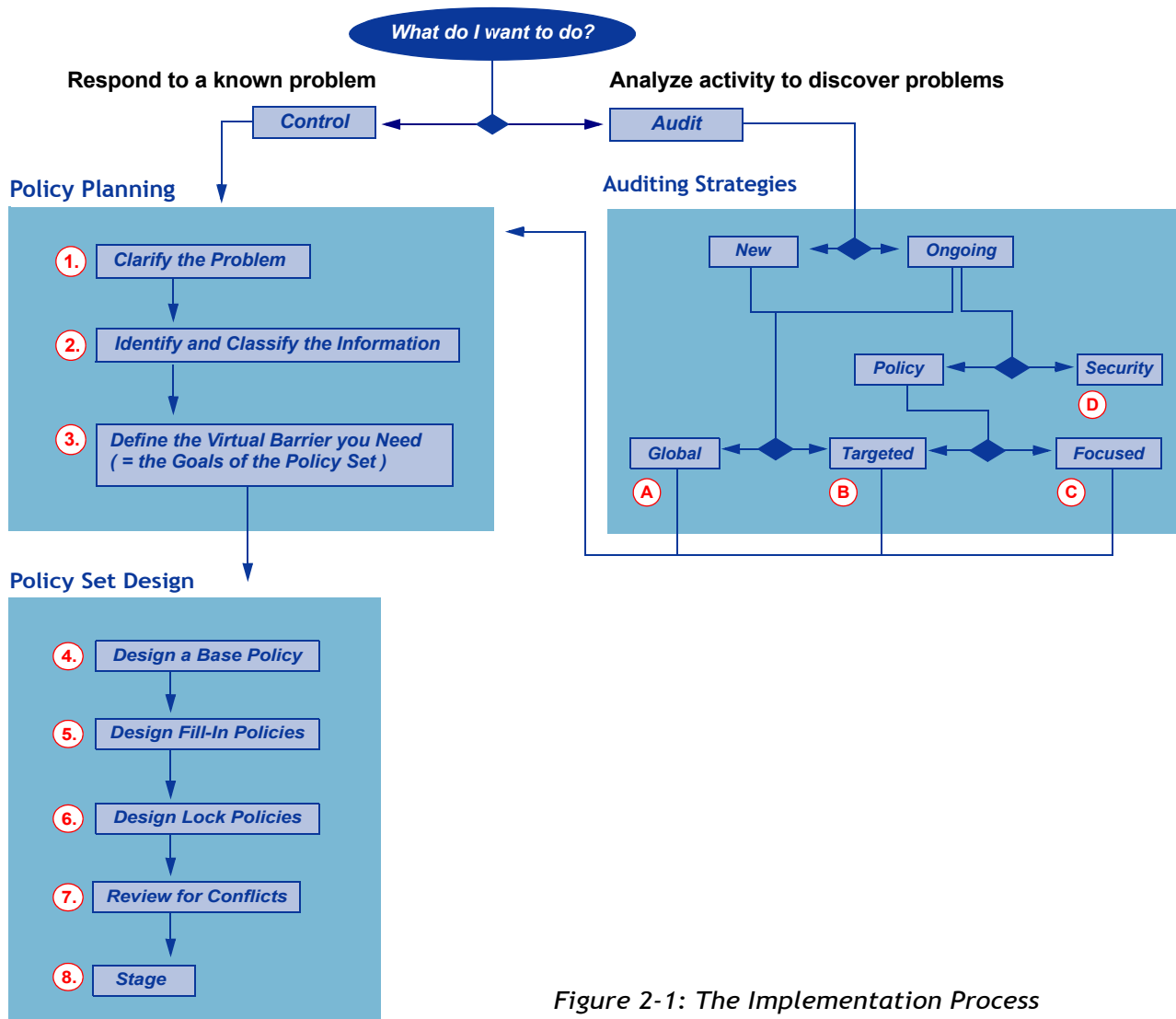


Figure 2-1: The Implementation Process

The Process

Broadly, any implementation of Compliant Enterprise will include the steps or phases listed below. Accordingly, these phases are reflected in the structure of the current manual.

1. Sizing Your System

Before you install the Compliant Enterprise software components, you will need to spend some time planning the number and size of the hardware hosts you will need, and how they should be distributed in your network to best meet your needs. Chapter 1 provides advice and help with this issue.

2. Designing System Audits

Once you have set up the hardware architecture and installed and configured Compliant Enterprise, your first step is to perform some analysis of how your organizations's information is being accessed, used, and distributed. In many cases an organization's executive staff or system administrators will have some idea of the information control needs, but in other cases they may have little or no idea. In either case, a systematic audit is sure to uncover useful insight into what kinds of control policies you need to design and implement. Compliant Enterprise provides a flexible set of tools for performing such information use audits at various levels; these are described in Chapter 3.

3. Designing Policies

The purpose of system audits is to expose the real nature of your organization's information control needs. Once you understand these clearly, you can proceed to designing sets of policies that work together to meet those needs. The policy design process is based on a number of considerations, including:

- the nature of the information you wish to control
- the specific business processes that involve this information
- the content and structure of your current Active Directory

it is a very simple matter to construct any policy you like in Policy Author, and then deploy it to your system. What is much more demanding, is the prerequisite process of policy planning and design, in which you carefully think out policies that will be most effective in meeting your needs, while also least intrusive into your normal business practices. Chapter 4 and Chapter 5 provide a detailed discussion of the policy planning and design process.

4. Staging and Testing Policies

Again, once you are ready to create your policies, it is very quick and simple to construct them in Policy Author. However, you should never deploy policies to your live production network without first staging them in some controlled environment where they can be observed and tested. This step is very important in uncovering unanticipated design flaws, policy conflicts, additional policy

requirements, and similar matters of fine-tuning. Advice on how to test and stage policies is provided in Chapter 6.

5. Monitoring System Performance

Once the policies are tested and refined, you can deploy them into your full production environment. Even though they have been staged and tested, it is possible that they may produce results you did not expect, or expose further information control requirements. For this reason, as well as for possible accounting or other internal purposes, you will want to monitor how your deployed policies are actually working in production, over time. The Compliant Enterprise Reporter tool allows you to do this very easily and effectively; complete information is available in the *Reporter 2.0 User's Guide*.

Information Activity Auditing

This chapter describes an additional powerful function of Compliant Enterprise, besides network information control: information activity auditing. This is often a preliminary step in gathering the information you need to start designing logical policies; but it can also be performed with the use of what we call *audit policies*.

The chapter is organized into the following sections:

- About Auditing
- Strategy A: Global Auditing ([page 20](#))
- Strategy B: Targeted Auditing ([page 22](#))
- Strategy C: Focused Auditing ([page 25](#))
- Strategy D: Security Auditing ([page 27](#))
- Ongoing Audits ([page 30](#))

About Auditing

Compliant Enterprise provides a set of powerful features for defining and enforcing policies governing how people in your organization access and use information resources. However, it also provides a benefit that may be just as valuable as information control: the ability to conduct comprehensive audits of how documents are being accessed and used throughout your organization, whether or not you are ready to implement any constraints on information users. How (or whether) you perform such audits depends on your organization's needs.

The Spectrum of Users

In theory, there is a broad spectrum of users who will find Compliant Enterprise extremely valuable. At one end of this spectrum are executives who have already identified their problem: they have a clear idea of what their information risks are, how information is being inappropriately used, and what control policies would help the situation. These users may not need to conduct an audit of any kind; they can start designing, constructing and deploying policies as soon as they finish setting up Compliant Enterprise.

Occupying the middle of the spectrum are those organizations who have a suspicion or a rough idea of how, where and by whom information is being misused, but need help in more precisely identifying the problems so they can formulate truly effective policies. These users may be able to define some helpful and effective policies without running an initial audit, but would definitely benefit

from an audit to help refine and focus their policies. In reality, most organizations probably find themselves in this middle ground.

And at the other end of the spectrum are organizations who may feel they have an information risk problem but have no clear idea what it is, or which users, documents or activities it involves; and without that, have little sense of how to design and implement suitable control policies. These users really must run an audit before proceeding to define any policies.

Auditing and the Active Control Solutions

Depending on how well you understand the nature of your enterprise's information control risks, you may know at the outset whether you plan to take advantage of the predefined solutions described in the next section of this manual, or not. If you do not, an information use audit is a good way of identifying the problems you face, and thereby recognizing if the solutions meet your needs.

If you do plan to use the solutions, bear in mind they will require some customization in every case. The predefined components in particular will need to be adjusted so they represent real entities in your environment. Much of this customizing will involve substituting generic entities in the component definitions with your actual entities: real LDAP user groups or user properties in user components, actual network directories or document properties in document components, and so on. This, in turn, may require that you identify rather precisely where information control risks are occurring, which users and groups they involve, and where the information in question should and should not circulate. For this too, an information control audit is a useful place to start.

On the other hand, some users can be confident they are ready to implement one of the solutions without any need for an initial audit. For example, you may simply know that you need to create a conflict of interest barrier to prevent any circulation of data between, say, your Mergers and Acquisitions group and your Equities Brokerage group. In clear-cut cases like this, there may be no need for an initial audit.

Auditing Strategies

The figure below illustrates the four basic strategies for auditing your network using Compliant Enterprise. Each horizontal double arrow represents a decision point leading to one strategy or another. The details on each of these points are presented below.

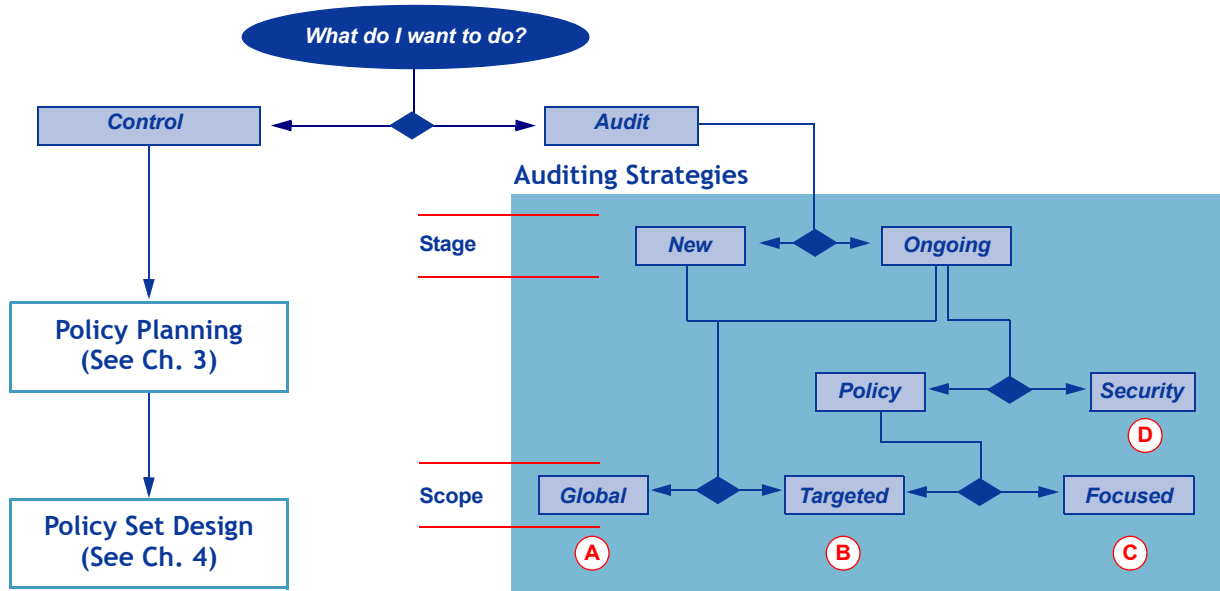


Figure 3-1: Four Basic Auditing Strategies

Stage and Scope

The first decision depends on what stage you are in your Compliant Enterprise implementation: whether you are running an initial audit, to help identify patterns of use in the course of identifying problems that need to be solved with information control policies; or have already deployed policies, and want to monitor how effectively they are serving the purposes for which they were designed. We refer to this distinction as *new* or *ongoing* audits. Since new audits are designed to discover patterns in a largely unknown information use environment, they will tend to have a broader scope than ongoing ones, as we will discuss below.

Strategy A: Global Auditing

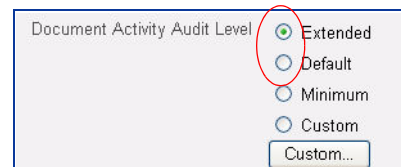
Compliant Enterprise allows you to approach audits in four general ways. For new audits, the simplest strategy is a *global audit*, which means simply turning the auditing feature on and capturing data on how all users in the system are accessing document resources, and what they are doing with them. You gather this information for a suitable amount of time—several days or a week, for example—and then use the Reporter tool to analyze the data. This approach will gather a large amount of data, but you can define report queries that let you easily sift through it to identify trends, anomalies, or concerns of any kind.

Global audits are very simple to implement, since they do not require that you define any policies or components, and because of their breadth they are well suited to new audits where you are trying to uncovering truly unknown problems. Of course, they are not technically limited to new audit contexts; you can run a global audit at any time, in order to catch newly emerging problems as they arise. However, bear in mind that they can be quite expensive in terms of processing resources and storage space, since they involve gathering and storing very large amounts of data—especially in large organizations, and when run for more than a few days.

Running a Global Audit

To perform a global audit, all you need to do after setting up your Compliant Enterprise system, is to flip a switch that starts the Activity Journal accumulating data on what every user is doing. Because no policies need be deployed, you can start this audit immediately. You can configure the operation to be entirely transparent, so that desktop users do not see any notifications and have no idea their activity is being monitored. (You can also enable the notification feature, if you prefer.) Depending on the size of your organization, you can run this for a day, a week or however long you wish, then run reports and analyze them.

There is one configuration requirement: in the default enforcer profile, you must change the Document Audit Activity Level from *Default* to *Extended*. This profile, automatically created on installation, is called *Desktop_Default_Profile*; you change the settings in Administrator.



During such an audit, the Activity Journal will gather a broad spectrum of information about information resource use throughout your network, including the names and locations of all documents or collaboration portal content each user is accessing, any actions each user performs on any document (copy, rename, save as, change properties, edit and then save, print, copy to removable media, or send via e-mail or IM); when the actions occurred, and what applications are used for handling all files.

Global Audit Reports

Because global audits gather such an indiscriminate collection of events, the planning comes not in gathering the information, but in defining the reports you use to extract what you want from the Activity Journal. You can define reports based on users, documents, sites or other portal content items, document loca-

tions, or times—or some combination of these. For example, you might examine data on one SharePoint library or site at a time, running an audit of how all content on it are being accessed; or you might focus on the file server for one organization in your company; or you might look at all file server access between 11:00 p.m. and 7:00 a.m.

You use Reporter's query definition screen to generate the reports you need. The top control on this screen allows you to select either Policy Activity or Document Activity; different filter fields are displayed depending on which is selected. For global audit reports, you need to select Document Activity. [Table 3-1](#) provides some suggestions for the way you can use the Document Activity query fields to filter for specific information.

Table 3-1: Queries for Global Audit Reports

| Query By | Examples |
|----------------------------|--|
| Individual users or groups | <ul style="list-style-type: none"> List all file server documents used by everyone in the Boston branch office. List all SharePoint sites used by regional sales managers. List all documents used by part-time contractors. |
| One or more actions | <ul style="list-style-type: none"> Show all cases when users sent e-mail attachments. Show all documents printed in the past week. Show all instances of moving a document from one SharePoint site to another. Show all cases of manually changing document-level security settings. Show all instances of cutting and pasting text from a specific set of documents. |
| Resources used | <ul style="list-style-type: none"> Show all users who copied, moved or cut & pasted files from the file server directory \SensitiveFiles\. Show all files used and operations performed on them by anyone using a shared-pool laptop. Show all files uploaded to any SharePoint site by members of the Special Projects Team. Show all instances when users tried and were blocked [or were allowed] to copy information to a removable device |
| Start and end dates | <ul style="list-style-type: none"> Give me all [whatever] that occurred since May 2. Give me all [whatever] that occurred over the weekend. Give me all [whatever] that occurred between 5 and 8 a.m. today. |
| Combination | <ul style="list-style-type: none"> List all documents e-mailed from company laptops by anyone in the finance department during the last two weeks of Q4. List all documents printed by contract employees in the graphics department. List all spreadsheets opened by any vice-president between 10 p.m. and 2 a.m. in the last week of the financial quarter for the last year. |

Limitations

As we mentioned, you can start running a global audit as soon as you install the Compliant Enterprise system and import your network information from Active Directory. However, in running reports filtered by users or groups, you will be limited to those that were defined in the Active Directory when you imported it. If you need more flexible, customized groups, you can use Policy Author to define them as components; they will then be available for queries. You are still not required to define any policies in Policy Author—just components.

It is very important to bear in mind that global audits can place a heavy load on system resources, and can generate very large amounts of Activity Journal data. For this reason, unless you truly have no idea what documents and activities you are concerned about protecting (and most organizations will have some idea), you will benefit from conducting your audit in a more targeted or focused way.

Also, note that there is a limit to the flexibility of the queries you can perform using only Reporter, since you can't define queries directly based on component properties. For example, you can't filter by directly by document type—e.g., “show all spreadsheets used by contractors”—or by time of day. (However, you can create a component based on the property, and filter on that.) This is generally not a problem, since the goal of the global audit is to look at everything that's going on precisely because you do not know what needs to be focused on. If you need a more finely grained query, you can run either a targeted or a focused audit.

Strategy B: Targeted Auditing

In a new audits you have the option of narrowing the amount of information Compliant Enterprise gathers, by restricting the audit in various ways. When you do this without writing and deploying any policies, we refer to it as a *targeted* audit strategy. A targeted audit can be restricted to a specified subset of hosts in the system, or to a specified set of actions that will be monitored and logged, or both.

What to Target

When thinking about how exactly you want to define your targeted audit, it is helpful to concentrate on the *information* you do know you are concerned about protecting—after all, most users in an organization will have some idea of which classes of documents are more sensitive than others. From this, you can identify classes of users who either need controlled access to those documents, or need to be restricted from using them. Recognizing both documents and users in this way is very useful in designing your targeted audit.

Another very helpful approach, related to the above, is to concentrate on the *origin of documents*. After all, many organizations deal with data repositories that are not document-based (CRM, HR, or accounting applications, for example), and so are not within Compliant Enterprise's direct control. However, such sources often have to generate documents at some point. That is, the circulation of application data is often a dominant concern, and in order to circulate data, some user has to use the application to generate documents. Thus by concentrating on the *application as a data source*, you can effectively identify broad classes of data that is likely to be sensitive, and define them as document components. (For a helpful example of this, see the Compensation Planning use case, as described in the “Sample Use Cases” chapter in the *Policy Author User's Guide*.)

For example, any Accounts Payable department uses some specific accounting program for its bookkeeping, such as Peachtree. While you cannot create policies to protect the accounting data directly, you can easily employ an applica-

tion component to prevent [any User using Peachtree] from [doing action you wish to monitor or block]. The same is true for any specialized CRM, help desk, inventory tracking, etc. applications; as well as for custom-built in-house applications, which can be an especially acute concern. This is one of the significant benefits of way Compliant Enterprise combines [users + applications + computers] as the unified subject of policies, as we will discuss in more detail (see [page 46](#)).

How to Target

Targeted audits are conducted using enforcer profiles. To begin, you define an enforcer profile that includes the restrictions you want. When you deploy the profile, the enforcers it applies to will gather auditing data only for the hosts and/or actions specified in the profile. This has the benefit of reducing the amount of data you are collecting. It is highly suitable for cases where, for whatever reason, you want to audit a known subset of computers in your network—one organizational team, for example, or one building on your campus—or audit only a specific kind of action.

Targeted audits are limited in that they can be based only on hosts, not actual users or user groups. Also, the actions they can target are limited to those listed below, and can't be refined by specifying properties or usage contexts.

- | | | |
|----------|---------------------------|-------------|
| • Read | • Change File Attributes | • Export |
| • Delete | • Change File Permissions | • Import |
| • Move | • Print | • Upload |
| • Copy | • Paste file | • Download |
| • Write | • Attach to E-mail | • Check in |
| • Rename | • Attach to IM | • Check out |

Like global audits, targeted audits also involve gathering information in the Activity Journal and then using Reporter to analyze it; but they focus on a defined subset of hosts and/or activities. You define this subset by creating an enforcer profile that includes only those actions you want to audit (in the Custom Journaling Options window), and associating the profile with only those hosts you want to audit. When the profile is deployed, only the specified subset of events will be written to the Activity Journal, which can then be used to generate reports as in a global audit.

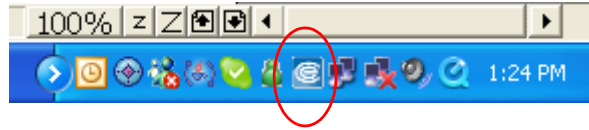
Targeted audits have the benefit of placing less load on system memory and storage space, and are useful in cases when you know want to audit only some specific hosts and/or activities.

Limitations

Targeted audits are limited in that they can only be defined based on the Custom Journaling activities, and on hosts. Also, as with global audits, you can define report filters based only on AD entities, rather than properties of those entities. To get to a more granular level of auditing, you need to define policies and run reports on them. We refer to this as *focused auditing*.

Using Silent Mode

By default, Desktop Enforcers run in *notification mode*, which means they display a CE logo in the Windows system tray and can pop up a notification message to inform the subject when policies are enforced. In most cases, this will likely be the behavior you want. If you choose, however, you can run agents in *silent mode*, which means the icon does not display, and the subject cannot use the right-click commands, such as View Notifications List. When policies are enforced in silent mode, the subject will see whatever standard Windows message most closely covers the case, but will not see any notification message even if one is defined in the policy.



You may want to consider using silent mode for both targeted and focused audits, since you are monitoring specific PCs for some reason, and may want the monitoring to be as transparent as possible.

Bear in mind this feature is available on a PC-by-PC basis (during installation), and not policy by policy. That is, each Desktop Enforcer is installed in one mode or the other, and all policy enforcement at that host will be in that mode. This means you generally make decisions about which machines should be in silent mode, rather than which policies.

However, to the extent that you can anticipate or plan which machines a given policy will be apply to, you can tailor policies that will be enforced silently. One way to do this, is to use computer components in your policy, referring to PCs where Desktop Enforcers are running in silent mode. You could even define a computer component that includes all the PCs where silent Desktop Enforcers are running; then you can use this component to design, in effect, silent-mode policies.

Strategy C: Focused Auditing

This is the most flexible and powerful auditing strategy. It involves defining policies that include logging obligations, which focus on the specific kind of activity you want to monitor. We refer to these as *auditing policies*; as a rule they will be Audit/Alert policies, since their purpose is not to prevent activity but simply to log it for auditing purposes.

Because focused auditing is based on policies, it lets you use all the components defined in Policy Author, combine them in intricate ways, and focus your audit using contextual factors such as component properties and time of day, week, or year. For example, you can define a policy that says:

Whenever any user in the Advanced Projects Design group copies any AutoCAD file from a protected location on the file server to a laptop, outside of normal business hours, log an event in the Activity Journal.

You can then use Reporter to produce a report on this specific policy, and easily identify all instances of the activity in question. This represents a far greater level of flexibility than either global or targeted audits offer.

Note that this is not the same as simply monitoring how many times your control policies are being enforced. (Though Compliant Enterprise also allows you to do that very easily, through Policy Reports.) Knowing that a policy or policy set is being enforced is not the same as knowing that it is effectively and comprehensively performing the functions it was designed for—in other words, that there are no control gaps left unfilled. A policy with a design flaw may be 100% enforced as written, but that does not mean your problem has been solved. Because focused audits are designed around the access activity itself rather than your policies, you can use them to test for any gaps or design flaws in currently deployed policies.

Running Focused Audits

As we mentioned, you run focused audits by defining policies, then generating reports on all instances when the policies were enforced. This has the benefit of letting you define very precisely the kinds of activity you want to track, based not only on broad component types (“show all documents opened by any user”) but on very narrowly defined properties and contexts (“show all documents larger than 10 MB, opened by non-full time users between 8 p.m. and 6 a.m.”). You define this level of granularity in the policies themselves, then simply track how and when the policies were enforced.

It is important to understand that by *enforced* we do not necessarily mean that any user was denied from performing some action, or even realized the action was being tracked; these should be Audit/Alert policies, whose only result is to write an entry in the Activity Journal whenever anyone performs the specified action. You can define such auditing policies as an analytical tool before you are, for whatever reason, ready to actually start blocking people from using specified information in specified ways. You can create any number of Audit/Alert policies, use them to gather your results for analysis, and then very easily

convert them to Deny policies, if you like, which will stop users from performing the specified actions.

Focused Audit Reports

Of course, once you gather the information based on enforcement of audit policies, you can use Reporter's query tools to generate reports, as we discussed above with global audits. However, because in this case the audit is specially defined according to the policies you deployed, your focus will naturally fall on those policies. For this reason, to run a targeted audit report you must select the Policy Activity search type.

As we mentioned earlier, Reporter's query definition screen displays a different set of filter fields depending on which search type you select, and the main difference—as [Figure 3-2](#) shows—is the two additional fields available for Policy Activity reports: Policy and Enforcements. For targeted audit reports, you will be most interested in reports based on these two filters, especially in cases when you have defined the policies specifically for auditing purposes.

The other difference between these two types is the list of actions available for filtering; Document Activity reports include the actions connected with Security Audits, and Policy Activity reports do not.

Use for Global Audit:

New Report

Query

Search: ☐ Policy Activity ☒ Document Activity

User:

Action:
 Agent Shutdown
 Agent Startup
 Change Attributes
 Change Security Settings

Resource:

Use for Focused Audit:

New Report

Query

Search: ☒ Policy Activity ☐ Document Activity

User:

Action:
 Change Security Settings
 Copy
 Create/Edit
 Cut & Paste

Resource:

Policy:

Enforcements: ☐ Allow ☐ Deny

Policy Activity Filters

Figure 3-2: Query Fields: Document Activity vs. Policy Activity Reports

Strategy D: Security Auditing

There are a number of system events that involve neither policies nor any kind of information resource access or use, but rather user activity. Monitoring this kind of activity is referred to as *security auditing*. It is important that you run security audits on an ongoing basis, since they may indicate attempts to circumvent policy enforcement. Security-related events include the following:

- Any enforcer shuts down normally
- Any enforcer shuts down abnormally
- Any enforcer restarts
- Any user tries to open an enforcer's configuration file
- Any user tries to open an enforcer's log file
- Any user tries to open an enforcer's binary file

Two other actions, User Login and User Logout, are also available and can sometimes be useful for security auditing purposes. As we mentioned, these eight security-related actions are available only for Document Activity reports, not Policy Activity (see [Figure 3-2](#), above).

These kinds of activities are always logged for all users in the system, which means that you do not even need to define a special enforcer profile to do it. A security audit really only consists of running a report query to extract the data from the Activity Journal, and examining the resulting report. You will probably want to define a few security report queries that can be run on a regular schedule. For example, one report might include all instances when any user tried to look at any enforcer configuration, log, or binary files; another might show all instances of normal or abnormal shutdowns and restarts; and another might give all user logins and logouts, which could easily then be sorted by user name.

It is also simple to monitor cases when any user either changes or is prevented from changing the security settings or the properties of individual files. Although this may be perfectly legitimate in many cases, it may also indicate efforts at circumventing policy controls, and you may want to consider including them in security audits.

Summary The table below summarizes the four initial auditing strategies we have presented in this chapter.

Table 3-2: Auditing Strategies: Summary

| Strategy | Description | Advantages | Disadvantages |
|--------------------|---|--|--|
| A. Global | Gather access and use data on all users, resources and hosts | Does not require policies Can be done as soon as CE is installed and configured Breadth allows it to identify problems you did not suspect | Generates large amounts of data to store and analyze Places heavy burden on system's processing resources Limited by the filters available in Reporter |
| B. Targeted | Gather access and use data on a target set of hosts and/or actions. These are specified in the enforcer profile, in Administrator. | Does not require policies Narrower definition requires less system overhead Suitable for cases where you have specific hosts or actions in mind | Can target only by host and action Limited by the filters available in Reporter |
| C. Focused | Define and deploy Audit/Alert policies, then generate Policy Activity Reports based on them | Allows you to focus on more granular criteria for the audit: can be based on any component you can define in Policy Author. | Requires more time and effort to define and deploy the policies and components |
| D. Security | Document Activity reports on the security-related actions: <ul style="list-style-type: none"> • Enforcer Stop or Restart • Abnormal Enforcer Shutdown • Access Enforcer Config Files, Log Files, or Binary Files | Does not require policies Run on events that are always logged Can defined and save the report once, the run it on a regular interval Should be run regardless of other kinds of auditing you are running | Narrowly defined set of actions Not relevant to policy enforcement |

Components, Policies, and Audits

The major benefit of running global reports is that you can do so without having to deploy any components or policies. The major benefit of deploying policies and components and then running focused audits, is the precision it offers in filtering on whatever entities or properties you like. However, there is a middle-ground option of defining and deploying components only, and then running a global audit. This requires less effort and time than focused auditing, while also broadening the usefulness and flexibility of the reports you can run, by offering more options for filtering. [Table 3-3](#) summarizes the benefits of running audits after deploying policies and components, components only, or neither.

Table 3-3: Components, Policies, and Global Audits

| Can Filter By? | No Policies or Components (Global Audit) | Only Components Deployed | Components and Policies Deployed (Focused Audit) |
|-----------------------|--|--------------------------|--|
| Action | Yes | Yes | Yes |
| User Name or Group | Yes | Yes | Yes |
| User Property | No | Yes* | Yes* |
| User Component | No | Yes | Yes |
| Document Path | Yes | Yes | Yes |
| Document Name | Yes | Yes | Yes |
| Document Property | No | Yes* | Yes* |
| Document Component | No | Yes | Yes |
| Application | No | Yes* | Yes* |
| Application Component | No | Yes | Yes* |
| Policy | No | No | Yes |
| Enforcement Effect | No | No | Yes |
| Access Time or Date | No | No | Yes** |
| Sites | No | No | Yes* |

*Create a component based on the property, application or site, then filter on the component.

**Create a policy based on date or time, then filter on the policy.

The basic differences are two: if you deploy policies you gain the ability to filter based on policies or enforcement effect (these are the distinguishing features of Policy Reports); and if you deploy components, you can filter based on them. Deploying components only is a useful middle-ground approach: you can do it without having to design and deploy policies, and it can provide a considerable auditing benefit.

Ongoing Audits

The above discussion presents audits in the context of an initial installation of Compliant Enterprise, as a tool for determining what policies to deploy. However, auditing is also an invaluable tool to use periodically after your policies have been deployed, both to measure how well the policies are performing over an extended time, and to help identify new information threats as soon as they emerge.

You will benefit by running both global and targeted audits on a regular basis—every week, for example—and analyzing various aspects of them, looking for anomalies and judging the effectiveness of deployed policies. The tools in the Reporter application allow you to focus in different ways. You can define queries based on overall information use (Document Activity reports), or on enforcement patterns for deployed policies (Policy Reports).

Lastly, it should be obvious that security audits should be run not only in the initial phase of your implementation, but also on a fairly frequent ongoing schedule—once a week at least.

As we discussed in Chapter 2, the process of creating and deploying policies may be divided into two basic phases. The first, more abstract phase involves clarifying the nature of your information control problem, and understanding what kind of information barriers would be required to solve it. We refer to this as the *policy planning* phase, and it is the subject of this chapter. The second phase, *policy design*, involves the more concrete formulation of policy sets that can create the information barrier you need. That phase will be covered in the following chapter.

This chapter is organized into the following sections:

- The Policy Planning Process ([page 31](#))
- Virtual Information Barriers ([page 33](#))
- Understanding Information Control Requirements ([page 36](#))

The Policy Planning Process

Users who are ready to begin implementing an information control regimen—that is, who do not need to conduct any system auditing, or have already completed it—must spend some time planning out their implementation before they can start designing policies. This planning is required in order to answer three fundamental questions:

1. What, exactly, does my information control problem consist of?

You need to ask this question in order to identify the documents and users that need to be involved in the set of policies you will eventually design.

2. How critical is the information at the heart of this problem?

Once you have identified the information, or classes of documents, you are concerned about, it is helpful to classify them according to their sensitivity or criticality. Once you do this, you will have a clearer idea of the control regimen you need to design—that is, how strict or thorough an information barrier you need to construct around that information.

3. What control barrier must I construct around this information in order to control it to a degree consistent with its sensitivity?

The answer to this question will involve document classes, types and groups of users, locations where documents may and may not circulate, types of applications that may be used with documents, and various contextual factors such as date or day of the week. Once you have identified these components of your information barrier in the ideal, you will be ready to translate them into policy sets in Compliant Enterprise.

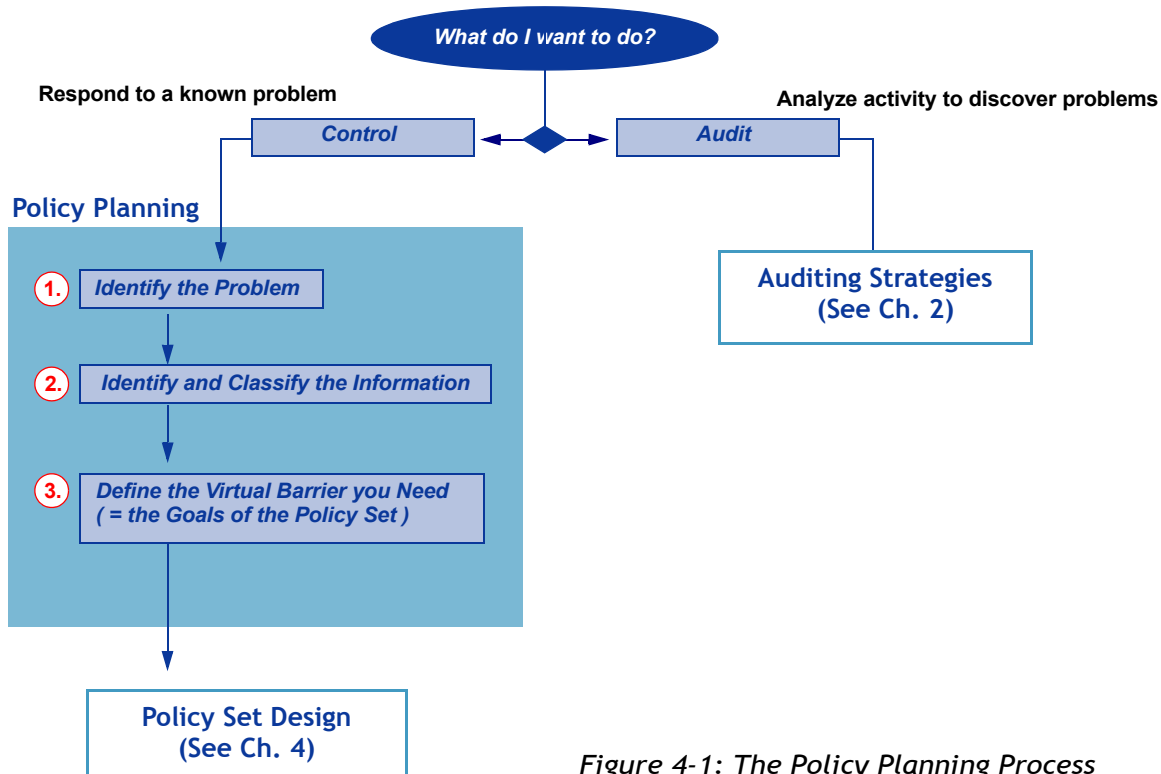


Figure 4-1: The Policy Planning Process

Before we turn to a closer look at these three questions, let's be sure we understand the concept we mentioned above—the *information barrier*.

Virtual Information Barriers

Thanks to rapidly evolving data storage and transmission technologies and shifting patterns of work and collaboration, the information control requirements of today's heavily information-dependent businesses and organizations are changing profoundly. Not so long ago, it was sufficient to define an enterprise network as Inside, then simply use firewalls and access lists to create a physical perimeter between it and the Outside. Now, unfortunately, there are all kinds of reasons why information has to be flexibly shared across this physical boundary, as well as restricted within it, in complex ways that often are far from compatible with traditional strategies of allowing or blocking physical access to network resources. [Figure 4-2](#) represents some of the challenges of this new situation.

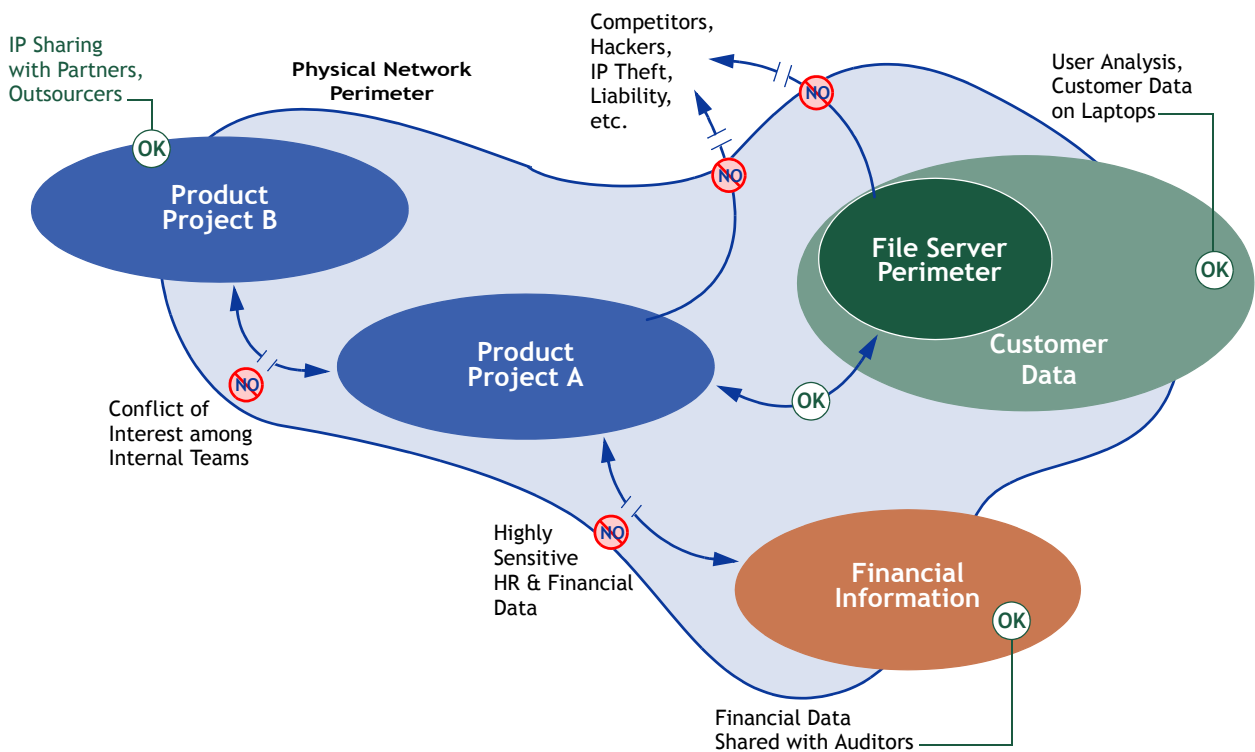


Figure 4-2: Virtual Information Barriers

Although there is technically one network perimeter around any single organization, the practical barriers you need to set up around discrete categories of information (represented by the ovals in the figure) will rarely if ever coincide with that network perimeter. That is, you may have classes of information wholly protected within the network, such as Project A, which must nonetheless be further isolated from various categories of internal users because of conflict of interest, or pure sensitivity considerations. Alternatively, you may have other classes of information that have to be used both inside and outside the network perimeter, either because they are needed by partners or auditors, or because

employees need to move the data around on mobile devices, or for various other reasons.

In short, the physical network barrier provides an inadequate definition both of how information may be traded within the organization, and of how information should be protected from being released outside. In both cases, it does not coincide with the practical requirements for business information management.

In addition to the traditional network boundary, what organizations need is a set of functional boundaries that can be flexibly defined based on the precise types of information involved, who needs to use it, and what they need to do with it. We refer to these as *virtual information barriers*. As the figure represents, they can exist within the network boundary or across it, but usually will not coincide with the whole network or even individual domains within it—they are defined functionally, and will change over time as business requirements evolve.

It is important to understand that this concept does not represent or even necessarily correspond to any physical or spatial barrier. Rather, it is a combination of all kinds of contextual factors that may or may not include location: the current role of the user, the point in the information's life cycle, the time of the day or day of the week, the software with which a user is may try to access the information, the medium by which he attempts to transfer it, and so on. Some combination of these and many other contextual factors may together define a barrier; uses consistent with the barrier's definition are permitted, and those that are not, are blocked.

The values of such a concept, if it could be implemented, are its fundamental flexibility and its capacity for definitional precision.

Enter Compliant Enterprise

The most useful way of thinking about the policies and components you create in Compliant Enterprise, is as tools for creating and maintaining virtual information barriers. Any users who know they need to implement some kind of information control regimen but aren't sure how to go about designing policies, will definitely benefit from thinking about virtual information barriers first.

These barriers are information-centric, meaning they are conceptually meaningful only in relation to some specific body of information that needs to be controlled. The term virtual information barrier may be defined as:

A set of practical limitations applied to a specific body of information, which constrains how it can be used, and by whom.

Just what the body of information is, in turn, derives from the concrete business problem you need to solve. Thus, the general approach in policy planning runs this way:

Problem -> Information -> Virtual Barrier -> Policies and Components

As [Figure 4-2](#) illustrates, you may need to define more than one barrier in an organization. Each barrier is designed to solve one discrete business problem, and is composed of several policies working together—a *policy*

set. You start with the business problem, focus on it to identify what specific information is involved, then define the nature of the virtual barrier that information requires, then design the policies that will create that barrier. This sequence is illustrated in the figure below.

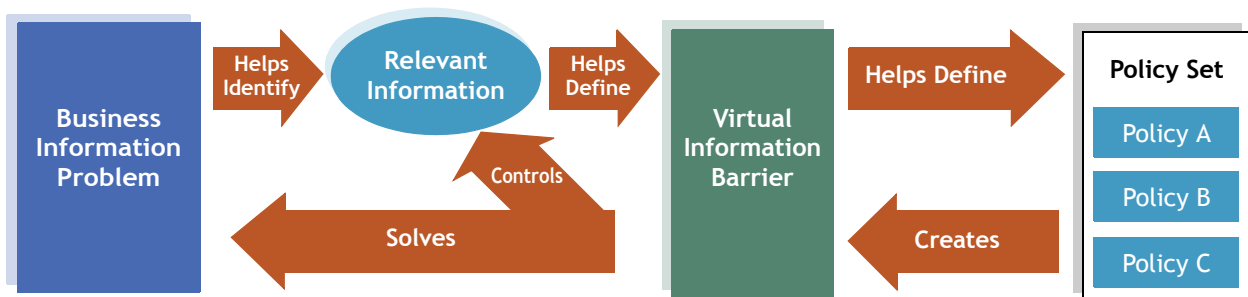


Figure 4-3: The Policy Implementation Sequence

Designing Barriers

It is important to understand that the core of this process is the information itself. After all, the problem you begin with is defined by the information that is involved: some kind of documents or data are being used in potentially harmful ways, which constitutes a problem you have to solve. The information also drives the definition of the barrier you need to construct, which is why we say that barriers are information-centric. This means that in defining your virtual information barriers, you must first identify the information you need to control. In doing this, it is helpful to sort it into *control categories*, as we will discuss in more detail below.

Once you have identified the information you are concerned about, you should then ask the following *barrier design questions*:

1. Who should be able to access it? And/or who must be prevented from accessing it?
2. From where should it be possible to access it?
3. To whom should it be possible to distribute it?
4. Where should it be duplicated and stored?
5. How should it be allowed to be manipulated or handled?
6. Is there a time context for any of the above? If so, when do any control requirements for it change, and how?

When you have considered and answered these questions, you will arrive at a conceptual definition of the boundary you need to solve the problem at hand. You will also have a helpful framework for designing the policies that will create that boundary. This is so because the answers to these questions easily map to the various parts of a Component Enterprise policy, like this:

1. Who = Allow Only Users / Deny Users; + application component (optional), + computer component (optional)
2. From where = Source document components (locations)
3. To whom = Actions, and target document components (i.e., network locations)
4. Where = Target document components
5. How = Actions in various contexts
6. When = Time context

These questions all pertain to the information at the heart of your problem. For this reason, you should focus on the information as the heart of your process.

Understanding Information Control Requirements

As we have seen, any Compliant Enterprise implementation should be designed to solve some concrete information control problem. The problem may be narrow in scope, as in a pilot program or a narrowly focused implementation, or it may be a comprehensive effort to regain control of information use throughout an entire organization. In any case, the success and ease of your implementation depends on how clearly you identify the problem you need to solve; and, as we have mentioned, the problem starts with understanding your information control requirements.

The process can be undertaken in the following four steps:

1. Identify the classes of information you need to control in order to solve the problem, and sort them into one of three *control categories*.
2. Identify a general set of *control guidelines* for each category. The controls you define for all classes of information within each category should conform to its control guidelines.
3. For each class, further define a set of individual control requirements that will ensure that the required control guidelines will be met for this specific class of information.
4. Analyze how closely those control requirements are being met in your current situation.

The goal here is to clarify how various information *should be* controlled, and then identify any gaps between that ideal and how it *actually is* being controlled. Once you identify these gaps, they will help you formulate the virtual boundaries you need to bring control regimens up to the required levels. Once you clearly formulate these boundaries, it will be much simpler to design policies that can implement them.

Clarify the Problem

You start by defining each individual problem you need to solve as precisely as possible. Here too, the process is “resource-centric”—that is, problems are defined by the class of information resources they involve. This will generally be

self-evident whenever you think about an information control problem, but sometimes you will need to refocus a problem in order to identify the document resources more precisely. For example, you may have a problem:

Our employees are carrying too much sensitive information around on their laptops or on USB thumb drives.

This problem may seem to involve users, documents and hardware devices, but for the purposes of designing a policy set, its focus should be on the documents. It is helpful to rephrase the problem to reflect this:

Sensitive information must be controlled so that it can circulate only in appropriate ways.

By focusing on the information, we naturally proceed to the next step: identifying precisely what constitutes sensitive information, and then determining what kind of circulation is appropriate for it. This involves thinking about degrees of sensitivity, and for this it is helpful to clarify some control categories.

Assigning Control Categories

The first step in this process is to analyze the information you are interested in, and assign it to control categories. (For our purposes here we will propose three categories, but you may choose to define more, and adjust the definitions to suit your particular circumstances.)

The criterion for categorizing a class of information is quite straightforward:

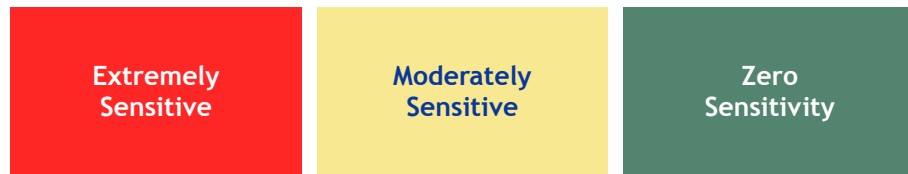
How much harm could be done to my organization if this information were released to anyone who wanted it, with no controls at all?

Bear in mind this is not a matter of current risk, or the degree to which the information is currently controlled or open (that consideration comes later), but rather of the potential harm it could cause if it were released. Obviously, different kinds of information occupy a spectrum, from completely harmless to potentially catastrophic. The weekly menu of your company cafeteria lies at one end of such a spectrum, while the blueprints for your gas centrifuge cascades lie at the other.

Control categories serve three main purposes. First, they help you formulate the virtual boundaries that you need to create around various classes of information. Second, they help you prioritize your policy design work: you should worry about defining and creating boundaries around more sensitive classes of information before you turn to protecting less sensitive ones. And third, they can help in weighing costs of implementing controls on different classes of information. (We will discuss this a bit later.)

Color Codes

One helpful approach is to use three color-coded categories: red, yellow, and green. These represent, respectively, extremely sensitive kinds of information, moderately sensitive kinds, and kinds that do not need to be controlled at all.



Although you will want to adopt standards relevant to your own needs, we can propose some more specific criteria for these three categories.

Red Information

Information is **Red** if its uncontrolled release could result in very serious financial burdens, significantly degraded competitive position, or clear damage to important interests of the organization. Examples of potential consequences of its release include the following:

- Liability for fines resulting from government regulations
- Critical loss of competitive advantage
- Gain of valuable confidential information by competitors
- High probability of lawsuits or other legal implications
- Serious damage to the organization's image or brand integrity
- The introduction of changes in the information that, if not caught, could lead to expensive losses. (Unauthorized changes to the formulas in a complex spreadsheet are an example of this.)

Yellow Information

Information is **Yellow** if its uncontrolled release could result in consequences that are undesirable and would better be avoided, but are relatively limited in their practical impact. These include:

- Embarrassment to the organization
- Remote potential of lawsuits or legal trouble
- Morale problems among members of the organization
- Gain of moderately useful information by competitors
- Possible loss of some competitive advantage
- Moderate harm to the organization's prestige and image

Green Information

Information is **Green** if its uncontrolled release would not produce any harmful consequences. Information in this category does not need to be controlled, and

for the most part falls outside the scope of your Compliant Enterprise implementation.

About Release

Note that by *release*, we do not specifically mean either inside or outside the organization; it might be either. The point is whether the information is under some kind of control, or not; or, whether it is controlled in certain ways, but frequently falls out of control in the normal course of its use. (This latter is the case with documents stored on a file server with access control; only authorized people can open and copy them, but once local copies have been made, they are essentially released.)

If information is not under reliable and continuous control, it has the potential to circulate anywhere, and the safest assumption is that it will. Put another way: unless information is controlled, you are not able to say with any confidence where it has circulated, and for planning purposes it is best to assume the worst.

Moreover, losing or exposing information outside the organization is not the only control risk; it can also involve potential for unauthorized editing by members of your organization, whether accidental or intentional, that can create serious and costly problems. Examples of this include accidentally deleting cells or changing formulas in complex spreadsheets, falsifying content in financial documents, or introducing regression bugs into product source code. Indeed, such changes even by trusted and authorized users can be a potentially huge problem, which is a major rationale for imposing a broad document use auditing regime rather than relying on trust.

A second point: control levels should not necessarily be equated with the degree of secrecy of documents—categories such as Top Secret, Classified, Confidential, and so on. A class of documents might be widely available to a large percentage of your organization’s members—that is, not particularly “secret”—but still be in the red category. The important point is not how many people have access to some information, but rather whether it should remain under control at all times, wherever it is circulated. Red information requires strictly controlled and auditable use, rather than (necessarily) narrowly restricted use. This is an important distinction.

One last, related point: control does not necessarily involve preventing anyone from doing anything with a given piece of information. In many cases it can mean simply being certain where a document is at all times, and knowing for certain who has used it and in what ways, after the fact.

The Time Factor

One other thing to consider, is that some kinds of information have a life cycle, during which its sensitivity can change. A given class of documents may belong to one control category at one point in its life cycle, and to another category at another point. For example, design sketches of a new high-tech widget that will be unveiled at an annual trade show may belong in the red category until the

day, hour and minute of the product launch, but become green thereafter. Another example is financial information before and after it is released in a quarter-end conference call.

For this reason, you may need to include a time context as you categorize classes of documents. This will help, for example, in distinguishing the policies you need to apply to design sketches before the big launch, from the policies applied after it. Usually this simply requires recognizing two distinct versions of a given class of information: in one relevant time context, and in another.

Defining Control Levels

Once you have identified appropriate control categories and assigned them to your information, you need to think about what specific level of control is required by each. Basically, for each category you ask:

Given the sensitivity of this category of information, what are the general guidelines for how it should be controlled?

This level of analysis yields broad rules for how any kind of information in a certain category must be controlled. For example, you may adopt the following guidelines for all Red information:

- Tight access control is essential. No one who does not truly need to use this kind of information should have access to it.
- When weighing security needs against any associated costs, security should come first. That is, you are willing to pay a high price to ensure the security of this class of information. (See "About Costs", below.)
- There should be no reliance on trusting individual users. The likely consequences of losing this kind of information are so harmful that it needs to be automatically protected regardless of the trustworthiness of authorized users—even senior management. (Laptops can be stolen or lost, after all, and paper documents can be fished from dumpsters.)
- People can always make mistakes; this kind of information must be handled in ways that are as mistake-proof as possible.
- Administrators should be able to know with complete confidence the audit history of this kind of information: who had access to it, who actually accessed it and when, and what they did with it.

Of course, this is still a rather broad set of guidelines, which will need further refinement. After all, there are many different classes of information in the Red category, which, while all very sensitive, need different control regimens because they are used in different ways, by different users, and with different frequencies. For example, you can allow source code to be handled in a different way than quarterly sales projections. Both, however, must conform to the broad requirements to which all Red data is subject.

Refining the Rules

To begin identifying more granular control regimens, you examine each class of information within a category. For example, let's say you have identified four

classes of Red information: quarterly sales forecasts, aggregate revenue figures before they have been released, new product design specifications, and sales contracts. You know that they all should be bound by the general controls over all Red information, but each needs more specifically designed controls. Toward this end, ask the following question of each of the four:

What can I allow users to do with this kind of information, and what must I prevent?

The results of this more granular analysis will constitute the proposed control requirements for each class—the ideal requirements for maintaining sufficient control over any information of that class, at all times.

Virtual Barriers

As you may already realize, this process of refining a second level of more granular rules is none other than defining the virtual information barrier that we discussed earlier. The overall question of what users can and cannot do actually encompasses the six barrier design questions we already introduced, but now you can pose them with specific reference to the benchmarks of their category's control requirements. For example, for a body of resources you have classified as Red, you would ask these questions in that context:

In order that this information meet the control requirements for all Red information:

1. Who should be able to access this information, and/or who must be prevented from accessing it?
2. From where should it be possible to access it?
3. To whom should it be possible to distribute it?
4. Where should it be duplicated and stored?
5. How should we allow users to handle it?
6. Is there a time context for any of the above? If so, when do any control requirements for it change, and how?

In answering all these questions, bear in mind the following considerations:

- The generic minimum requirements, dictated by the color category.
- The way this individual class of information is used, including such factors as:
 - how much it really needs to move around, in the normal operation of your organization
 - how many people actually need to use it, ditto
 - whether its sensitivity changes over time
 - whether it has any intrinsic characteristics that effect how it can be used (e.g. the size of file format, or a specialized application required to use it)

- The costs that would be associated with controlling access or use in various ways.

In reviewing these considerations, you may expose contradictions between the ideal control requirements of a color class, and your organization's need for access, flexibility and mobility. Exposing them explicitly this way will help to resolve them, either by adjusting the categorization, or by accepting that access must be more constrained than previously thought.

To some extent, this tension is intrinsic to all information control problems. After all, every information control problem involves finding the most appropriate point along a spectrum from complete openness to total lock-down. This is complicated by the fact that every point on the spectrum has not only security benefits, but also costs.

About Costs

There is always a trade-off between the benefits of controlling information, and certain costs that such control imposes. Prominent among these costs is the potential for interfering in the work flow of an organization—for example, by preventing users from handling or transferring information in a way they are accustomed to (taking work home on their laptops, say), or by requiring that certain classes of data be stored only in one specified location and nowhere else. Another kind of cost is the annoyance or resentment of users who are blocked from opening certain kinds of documents, but who feel they should have access. Another is the aggravation imposed upon exception cases—people who really do need access to some class of information, but have been improperly blocked due to design flaws in a policy. Still another, more obvious cost is simply the time required to design and test a policy, and to field the inquiries that may arise from users on whom the policy is enforced.

Considering costs is a matter of balance: whether they are outweighed by the benefits of information control. Of course, costs should be kept to a minimum by designing policies carefully, so as not to impose unnecessary restrictions on work flow, or include users or information erroneously. Still, some policies may be intended to change the work flow in order to impose adequate information security, and users will simply have to get used to the change. Similarly, the inconvenience of users who are no longer allowed to take certain documents home may simply be an appropriate price for your required level of security. In both cases, the policy designer has to anticipate the costs and weigh them against the expected benefits, and make a choice.

One of the purposes of identifying control categories, is that they can help in making decisions about cost. For example, a certain level of cost may be acceptable in a policy that protects a type of red information, but may not be acceptable if the policy applies to yellow information.

Measuring Up

Once you have identified categories of information and classes within them, and formulated the ideal virtual boundary for each class, you can then compare

each ideal with the way documents in that class are actually handled in your organization. The gap between these two is what you need to close.

In many cases the gap may be obvious; you may have no control at all over certain kinds of spreadsheets, say, other than a simple access control list on a file server. In other cases—source code that is maintained in a sophisticated version control system, for instance—your current situation may come fairly close to the ideal, giving you a relatively small gap to fill with Compliant Enterprise. Whatever the case, identifying your ideal virtual boundary in this systematic way ensures that the policies you define will be appropriately and effectively targeted to your needs.

About Network Audits

This matter of comparing current use patterns to an ideal is complicated by the fact that it is often hard to know who does have access to what information, and what authorized users are doing with it. Any network will have at least a basic system of access lists controlling hosts and directories on file servers, but in large organizations where people change their roles and move among access groups over time, permissions are very hard to maintain or even keep track of, and reliable information on who is doing what is difficult to gather.

And yet, such information is quite important in designing effective policies. Some unknown number of users may be opening and distributing files they shouldn't, but you can't design a Deny policy unless you know who they are. It is easy enough to block most users with an Allow Only policy, but you need some confidence that the policy includes everyone who needs to use the documents in question, and blocks only people who do not. That is, unless you know who all the users are who do need authorization, or what they should be allowed and not allowed to do with a given class of information, then it is hard to formulate the appropriate Allow Only policy.

To help with this, you can use Compliant Enterprise run a network audit, which may be as broad or narrow as you like. You can gather data on how all users on all monitored PCs are using all information resources; or you can narrow the target to certain specified PCs, by user, but still capturing resource use; or you can focus on individual users and specific classes of information. Once Compliant Enterprise gathers this information—over the course of a few days or a week, say—you can then generate detailed reports, which you can filter by user, host, application, time of day, and so on.

For a more detailed discussion of this powerful tool, refer to Chapter 3.

Moving On

Once you have identified and classified the information you are concerned about, conceived of the ideal virtual barrier you would need to protect it appropriately, and come up with a definite set of rules that would provide such a barrier, you are ready to translate these rules into a policy set. This is the subject of the following chapter.

In the previous chapter, we suggested a systematic approach to the policy planning phase, which involves recognizing the business problem you want to solve, identifying and categorizing the information the problem arises from, and defining the virtual information barrier that would offer a solution to the problem. In this chapter, we turn to the next step: designing the set of policies that will create the information barrier you need.

The chapter is organized into the following sections:

- Components and Policies
- What You Can Do ([page 49](#))
- About Policy Sets ([page 58](#))
- Guidelines for Design ([page 68](#))
- Policy Set Examples ([page 71](#))

Components and Policies

In Compliant Enterprise terminology, *components* represent any entities in your organization—workers, documents, laptop PCs, file servers, portal sites, and so forth—that play a role in your information control rules. They are the building blocks of *policies*: you use Policy Author to create them and to combine them into coherent rules that combine to create your virtual information barriers. Components can represent entities, such as people, computers, or documents; or actions, such as opening, copying, or deleting. The way you combine them is rather like the way you use parts of speech in a sentence, which is why we refer to the *grammar* of policies.

For example, you may want to prevent any contract employees from attaching engineering specs to outgoing e-mails. That goal constitutes a logical policy. To express it, you need two object components—*contractors* and *engineering specs*—and one action component—*attach to e-mail*. You then combine these three components in standard grammatical syntax, subject-verb-object:

[contractors] - [may not attach to e-mail] - [engineering specs]

Defining components is necessarily closely related to, and to some degree constrained by, the tools that Compliant Enterprise provides. For this reason, the policy set design process must be based much more firmly than policy planning is, on an understanding of what you can and cannot do in Compliant Enterprise.

Typical Policy Goals

Here are some examples of policies you can implement in Compliant Enterprise.

- Prevent two groups of users from accessing each other's resources, either on file servers or collaboration portals
- Prevent duplication of a class of documents to less secure locations
- Prevent modification of a set of finalized documents
- Prevent e-mail distribution of a set of documents
- Notify a compliance officer when a particular document is accessed or modified
- Limit the types of applications that can be used to create, modify, or read documents
- Limit the systems from which a document can be accessed
- Control the systems to which a document can be copied or moved
- Prevent unauthorized printing of a particular document
- Prevent modification to a document after a certain date
- Prevent modification of a system configuration file outside a certain time window
- Log modification of security attributes for particular documents or folders and notify the appropriate person of the modification
- Prevent the duplication of information from a document

Policy Design

Policy design is the process of creating policies and policy sets, and specifying which components are involved and what consequences will occur when users attempt to perform specified actions. Policy design involves putting the appropriate policy components together along with a predefined set of possible actions and conditional expressions. For example, an organization might develop a policy that allows people on the graphic design staff to create, edit and save Flash files, but denies all others permission to do so.

Policies are based on the Active Control Policy Language (or ACPL), with a strict grammar that dictates the structure of each policy. Every policy is made up of the following:

- The **subject**, such as a particular set of users, to whom the policy applies. Subjects are represented by user components defined during the modeling process. Optionally, users can be constrained by including one or more computer components and, also optionally, one or more application components, as shown in [Figure 5-1](#). Note that if you add one or both of these optional components, the logical subject is *all of them combined*, and the policy will apply only in cases when all of them are present.

For example, a policy that prevents *sales reps using notebook PCs* from mailing a certain class of files will not prevent them from mailing those files from a desktop PC, since *sales reps using PCs* is a different subject, not covered. That is, the subject consists not only of

the user component, but of the user and computer (if any) and application (if any) components taken together. This is an important point to consider as you define policies that involve computers and applications.

- The **action**, such as trying to open or copy a file, about which the policy is concerned. As we have seen, each action component may consist of one or more than one basic actions. In addition, you can include more than one action component in each policy.
- The **resource**, which describes some category of files, to be covered by the policy. Resources are represented by document components, defined during the modeling process.

Optionally, the policy can include additional context-based conditions that must be met in order for the policy to be enforced; for example, a policy might cover only files created before a certain date.

The policy then specifies the enforcement effect: whether to allow or deny the action.

Figure 5-1, below, illustrates the way components can be combined to define policies.

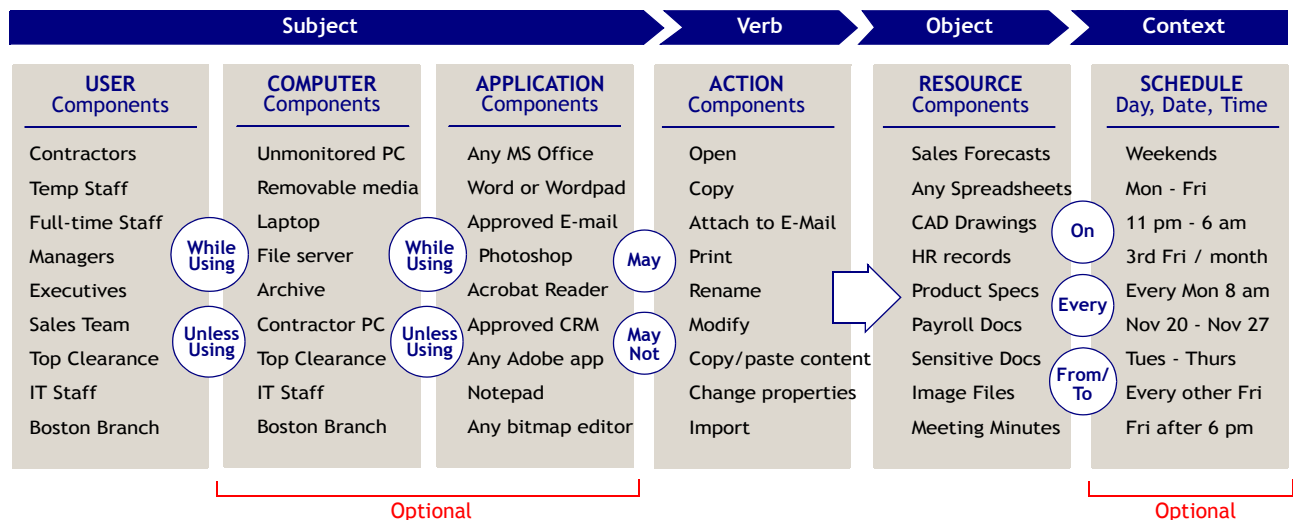


Figure 5-1: Combining Components

A policy also specifies any additional follow-up tasks to be done after the policy is evaluated, such as adding a record of this policy event to the log file, or sending a notification e-mail to a particular person. In Compliant Enterprise terminology, these follow-up tasks are referred to as *obligations*. The obligations currently supported include:

- Log this action whenever it is permitted by the policy

- Send an e-mail notification whenever this action is permitted by the policy
- Log this action whenever it is restricted by the policy
- Send an e-mail notification whenever this action is restricted by the policy
- Log this action every time this policy is invoked, regardless of whether it permits or restricts the action
- Send an e-mail notification every time this policy is invoked, regardless of whether it permits or restricts the action
- Perform some custom-built obligation. Custom obligations can involve any action that can be performed by an executable file, which can be invoked when the policy is enforced. Examples include encrypting files whenever they are copied to a certain location, prompting users to submit a special acknowledgement before they can access certain information,

Design Sequence

Note that although components are logically the building blocks of policies, this does not necessarily mean that you will define components first and then design policies second. Practically speaking, it is often easier to identify the policies you need to implement and then, based on those, extract a list of the components they require. In other words, there is no need for any sequential priority between the two: as you think about policies you will simultaneously identify components, and as your stock of components grows it will help clarify how they can be combined into useful policies.

Obligations are a somewhat different matter. You may find it convenient to think about them after the components and policies are defined, as a separate consideration. Even so, in the course of thinking about obligations, you may come up with additional policies designed solely for the purposes of monitoring and logging activity on the network, without restricting any use.

Bear in mind that obligations—a pop-up message to the policy subject, or an e-mail to an administrator with out the subject's knowledge, or both—can often be the sole reason for defining a policy, even if it does not prevent any action. Such policies, called *Audit/Alert policies*, can be useful for investigating and identifying suspicious behavior, or for simply reminding users of information restrictions, upcoming deadlines, or other matters.

What You Can Do

In this section we will present a bit more detail on the kinds of actions and objects you can represent as components, the ways you can combine components into policies, and the context-based conditions you can attach to those policies. This is important to understand even for users who will be designing policies, since anything you design obviously must be consistent with the capabilities of Compliant Enterprise's Policy Author tool.

Components

There are six basic types of components:

- Actions
- Users
- Computers
- Documents
- Applications
- Portal Content

One of these describes actions; the rest represent objects. Let's discuss each type in turn.

Action Components

The action components you define actually comprise one or more *basic actions*. Basic actions are divided into three functional categories:

Table 5-1: Basic Actions

| Category | Basic Actions |
|------------|---|
| Access | Read, Rename, Write/Modify, Delete, Change Attributes, Change Permissions |
| Transform | Paste, Move, Copy/Embed |
| Distribute | Attach to E-mail, Attach to E-mail IM, Print |

You cannot define any new basic actions, but you can combine them together, within action components. When they are combined, a logical OR is assumed. For example, you might define a component called Distribute, that covers e-mailing, attaching to IM, printing, copying, or moving a document. This feature provides considerable power and flexibility in targeting components to the specific needs of your policies. This is an example of how thinking about policies first often helps identify the specific components you will need to create to implement them.

Object Components

You can define the other types of components—users, computers, documents, applications, and portal content—based on their properties. Naturally, these will differ depending on the object type. For Users and Computers you can customize the available properties, based on whatever fields are defined in your Active

Directory. For all four types, the following default properties are available in Compliant Enterprise.

Table 5-2: Default Properties, Object Components

| Type | Default Properties |
|----------------|--|
| Users | Account Name, Company, Country Name, Department Name, First Name, Full Name, ISO Country Code, Last Name, Numeric Country Code, Title, User Principal Name |
| Computers | DNS Host, Machine Name, Operating System, Network Address, Site |
| Documents | Access Date, Created Date, Modified Date, Name, Type, Directory, Owner, Owner User Component, Owner LDAP Group, Size Note that you can also define components that represent whole directories, rather than the files in directories. These are very useful for preventing users from circumventing document-based policies by moving, renaming, deleting, or otherwise manipulating parent directories. For details, see “Directory Components” (page 55). |
| Applications | Name |
| Portal Content | Resource Signature, Name, Title, Date Created, Created by, Date Modified, Modified by, File Size, Description |

Policies The basic structure of a policy requires an effect, an object component for the policy subject (usually a User), an action, and an object component for the policy target (usually a Document). For example,

```
[Deny] + [Contractors] + [from Copying] + [Spreadsheets]
```

In addition, there can be one or more other elements that define the specific context in which the action takes place. These can be provided by Computer components that modify and restrict the subject of the policy:

```
[On Company Computers]
[Deny] + [Contractors] ^ [from Copying] + [Spreadsheets]
```

You can also use Application components in this way:

```
[using anything but MS Excel]
[Deny] + [Contractors] ^ [from Opening] + [Spreadsheets]
```

Lastly, you can always refine a policy's context by adding a time component, which we also refer to as a *time trigger*:

[between 8 p.m Friday and 7 a.m. Monday]

[Deny] + [Contractors] + [from Opening] + [Spreadsheets] ^

Choosing a Policy Effect

The basic point of a policy is to restrict some users from performing some action under some specified conditions. (*Audit/Alert* policies are the exception to this; they are designed only to log events, without restricting users.) Bear in mind you have two ways of restricting actions: you can prevent a category of users from doing some action, while allowing all others; or you can Allow Only some category of users to use some category of documents, while preventing all others. This is referred to as a policy's *effect*. Choosing the effect you want—Allow Only, or Deny—is the starting point for every policy.

As we have mentioned, as a rule the base policy of a given set will be Allow Only, and fill-in policies will be Deny. However, this is not an absolute rule, and you may find cases when it makes sense to break it. Which effect you use for any policy depends on the logical nature of the policy, including the following considerations:

- whether it will restrict a small or a large number of users
- whether it is more practical to identify the users who are allowed or the users who are denied
- whether allowing or denying is the exceptional case, as opposed to the norm

Operators

All components you include in a policy are governed by a logical operator, typically the equivalent of *Is* or *Is Not*. This provides additional flexibility in designing policies based on exclusion logic—"anything other than" expressions. These will generally be used in Deny policies. For example, you might want to deny users from sending attachments using anything but your corporate approved e-mail applications, or from copying a document to any location other than the designated folder on a secure server.

Logical operators have the effect of reversing a clause's logic, just as Deny/Allow Only reverses the effect of the whole policy. This means there is nearly always more than one way to express the same intent; "Deny all new hires", for example, is logically equivalent to "Allow Only all Non New Hires".

The operators available at the lower, component clause level, allow you to use multiple negatives to design very narrowly targeted policies. However, they can become confusing. As a rule, it is important to keep your policy logic as clear as

possible, and one way to do this is by expressing policies in positive terms rather than negative, whenever you have an option.

Nested Properties

You can combine as many properties as you need to define the component or policy you have in mind. You can use And or Or operators, or both together—for instance, “all documents in Directory X or Directory Y, that have a file size above 5 MB and an .MPG extension.” This also allows you to create very elaborate and precise components, if you need to.

Your ability to use some of these properties, especially of documents, may depend on standard practices such as names assigned to types of files, and locations where certain types of files are stored. However, you can also use Compliant Enterprise to help enforce some of these structural prerequisites. That is, you may need to define one or more policies as requirements for ensuring the proper operation of other policies. For example, you can Deny all users from saving any document whose name does not conform to the standard for the required directory location.

Use Wildcard Characters

Wildcard characters can be very helpful with this kind of conformance enforcement policy. For example, you may have a company rule that no social security numbers may be used in file names. This can be easily enforced by the policy:

```
FOR *###-###-###*. * OR *#####. *  
  
BY ALL USERS  
  
ON SAVE AS, CREATE NEW  
  
DO DENY ALL  
  
OTHERWISE DENY
```

Control Use Schedules

Policies can include time triggers that can be used to enforce information use according to a schedule. Triggers can be expressed as dates, days of the week, and times of day. Time triggers are very useful for enforcing access to information during certain times, such as outside standard work hours, or within certain windows such as the end of the financial quarter. They tend to work better in contexts when strict controls are needed, and are less suitable for others where more flexibility is required.

Bear in mind that you can define Audit/Alert policies that include schedules, which can serve as reminders about regularly occurring deadlines, or other workflow schedule requirements. (Audit/Alert policies can display a notification without preventing subjects from performing some specified action.) For example, whenever a contract employee opens a document on the second or fourth

Thursday of the month, that could invoke a policy that doesn't block any action, but just displays a reminder that time sheets are due tomorrow.

Create Lock Boxes

You can define document components that represent network locations where, whenever a document is placed there, it is governed by whatever restrictions you like. In effect, this can be like dropping documents into a lock-box: someone can create a draft document and place it in the specified location, where he no longer has access to it himself. This can be used in combination with use schedules, so that various lock-box directories are enabled on certain days or for part of the month, and then opened up at other times.

You can also define policies that help enforce the use of such lock boxes by combining them with "reverse lock-box" policies: preventing specified users from using documents in certain ways *anywhere except* the location where they should be stored. These can incorporate use schedules to help enforce the proper handling of documents not just in space, but over time as well.

Example

Here is an example of several policies that would work together to control workflow—in this case, the handling of sales forecasts.

Let's say that all WidgetCo. Inc.'s sales representatives must use an approved spreadsheet template for their biweekly sales forecasts. They have to submit their forecast numbers to their regional manager every other week. The regional manager then reviews and collates them into a single spreadsheet, which he uses to track longer-term expectations. Shortly before the end of each quarter, the regional manager submits his consolidated numbers to the Sales VP. The VP then reviews the projections, adjusts them if necessary, approves them, and incorporates them into his broader revenue projections.

You can use lock-box document components, each with a strictly defined timer, to enforce that users submit documents only to a specified correct location, and only during the required time window. These components would work as one user's outbox and another's inbox; as soon as one user drops a document into the outbox he loses access and write control, the recipient gains access, and the workflow is automatically enforced. When the workflow cycle is completed, the normal permissions are automatically restored.

General Considerations

Compliant Enterprise offers several features that are available for special cases, which you may want to use in some cases and not others.

Custom Obligations

Custom obligations provide a very powerful tool for extending how you can benefit from Compliant Enterprise. Custom obligations may be any kind of executable code that can be triggered by the enforcement of a policy. You can create them as independent executable files, which can then be specified in the Obli-

gation Name field of any policy where you want to use them. When the policy is enforced, the executable is invoked and the resulting behavior occurs.

Here are some examples of custom obligations:

- **User Prompt:** A policy prevents some user from accessing the files in a directory, then pops up a web page describing the security policy that is involved, and offering the user an e-mail link to request access if he think the policy has been incorrectly applied.
- **Removable Media Encryption:** A policy allows some users to copy sensitive data to Removable Media, but automatically encrypts the data before copying.
- **Call Pager:** A policy denies some users access to project files, and sends a call to the project owner's pager to notify him of any attempts to open the files.
- **Scrub Lost Laptop:** A policy dictates that all non-system files are automatically deleted whenever any user presses any key. This policy can be deployed as soon as a laptop, for example, is determined to be lost or stolen. As soon as any user connects the laptop to the Internet the policy is deployed, and comes into effect.

The possibilities for using this feature are extremely broad, since any action or behavior that can be expressed as an executable file can be incorporated into a policy as a custom obligation. Complete details on creating, configuring and using custom obligations are provided in the *Compliant Enterprise 2.0 Administrator's Guide*.

Using Silent Mode

By default, Desktop Enforcers run in *notification mode*, which means they display a CE logo in the Window system tray and pop up a notification message to inform the subject whenever a policy is enforced. In most cases, this will likely be the behavior you want. If you choose, however, you can run enforcers in *silent mode*, which means the icon does not display, and the subject cannot use the right-click commands, such as View Notifications List. When policies are enforced in silent mode, the subject will see whatever standard Windows message most closely covers the case, but will not see the notification message even if one is defined in the policy.

Bear in mind this feature is available on a PC-by-PC basis (during installation), and not policy-by-policy. That is, each enforcer is installed in one mode or the other, and all policy enforcement at that host will be in that mode. This means you generally make decisions about which machines should be in silent mode, rather than which policies.

However, to the extent that you can anticipate or plan which machines a given policy will be apply to, you can in effect tailor policies that will be enforced in silent. One way to do this, is to use computer components in your policy, referring to PCs where enforcers are running in silent mode. You could even define a

computer component that includes all the PCs where silent enforcers are running; then you can use this component to design, in effect, silent-mode policies.

“Directory Components”

When you define a Document component, you have the option of assigning it a special property called *Include Only Directories*. When this property is set to Yes for a component, the component will only apply to any directories that are consistent with the definition of the component, not to the files those directories contain. Practically speaking, this enables policy designers to incorporate handling of whole directories in their information control strategies. This feature is particularly useful in creating lock policies.

For example, if you construct a policy that prevents certain users from copying the files in a specified file server directory, based on a document component describing the files in that directory, the policy will not stop a user from copying the entire directory, or its parent directory—thereby evading the control of the policy. In order to avoid this problem, you can design a second component with the *Include Only Directories* property enabled, and include that component in a policy that applies the same controls to the parent folder, as to the files in the folder. This will mean that the policy for the file-based component will protect the individual files, and the policy for the parent Directory-based component will protect the parent directory.

Note that this strategy can apply to many levels of a directory structure—meaning that you do not need a policy to protect every directory specifically, but only the topmost one (defined with the **** wildcard characters) in a directory tree containing sensitive documents. This means the relationship of folder policies to file policies need not be one-to-one, but rather one-to-many: you do not need one folder policy for every file policy, but only one **** folder policy that can protect all files in all the downstream directories.

For example, say you have a file server with a directory called *Financial*. You need to control the Excel spreadsheets in two directories, *Financial* itself, and *Financial\Forecasts\Revenues\2007*. To do this, you would need three policies: one each to cover the files in the two directories you are concerned about (1 and 2), and another (3) to cover the directories themselves, starting with the highest-level directory required. These policies would look like the following:

Policy 1: Allow only [USERS] to access document component c:\Financial*.xls

Policy 2: Allow only [USERS] to access document component c:\Financial\Forecasts\Revenues\2007*.xls

Policy 3: Allow Only [USERS] to Move, Delete, Copy, Rename document component c:\Financial** (*Include Only Directories* = Y)

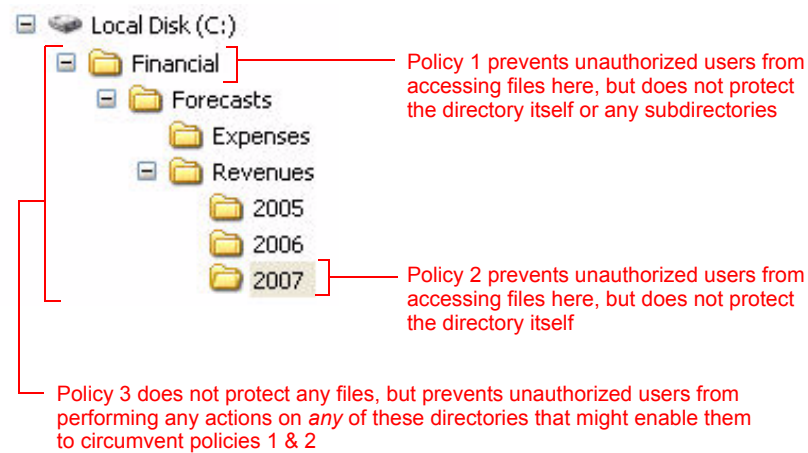


Figure 5-2: Using “Directory Components”

The basic strategy for directory components is that you need to deploy at least one of them to control the directories containing sensitive files, but the relationship will generally be one-to-many: one directory component can suffice to cover the policies for files in all downstream subdirectories.

Using Active Directory Synchronization

Today’s networks tend to be highly dynamic, with users, computers, documents, and other entities being added and removed frequently. Because it is important that the Information Resource Model in Compliant Enterprise stays in sync with these changes, you can use the synchronize utility to update it automatically, on a set schedule. This utility re-imports the contents of your Active Directory into Compliant Enterprise, to ensure that all deployed policies properly cover newly introduced entities.

This feature has a subtle implication for policy design, since some kinds of components are synchronized automatically and others are not. Specifically, because information on users, user groups, computers and computer groups is stored in Active Directory, any changes to them will automatically be reflected by Compliant Enterprise components of that same type. For example, if Jane Jones moves from the Marketing department to Sales, her new position will be stored in her record as a user in AD. If she is part of a Marketing Personnel user component in Compliant Enterprise (“all users whose Dept. = Marketing”), the synchronization will automatically remove her from it, and this will be reflected in the operation of all deployed policies that use that component: she will lose access to the Marketing Projects directory on a protected server, say, while gaining access to a Sales Forecasts directory. This will happen automatically, without requiring any manual changes to either components or policies—obviously an advantage in terms of scalability and maintenance overhead.

However, this benefit only applies to the components that are reflected in AD entities. Those that are not will be less flexible in the face of network changes, and may require more manual adjustment from a Compliant Enterprise administrator. For example, if you define a component based on too granular a property—a user component consisting of several individual user names, for instance: “Marketing ad hoc committee = Jane Jones, Mike Chan, and Tanya Baker”—then any changes for that component in the network will not automatically be reflected in deployed policies.

The general guideline, then, is to bear this issue of autosynchronizing in mind as you design policies and components. Whenever possible, you should avoid designing policies or components that will require manual updating to reflect changes because they cannot be updated by an AD synchronization. As a rule, you can do this by referring to groups or classes of entities as defined in AD, rather than individual entities.

Policies relying on sites will not be autosynchronized, since sites are defined arbitrarily outside of Active Directory. For example, you may define two sites for your company, representing Building One and Building Two on your campus, but if an employee happens to move from one building to the other, the information in AD may not be sufficient to automatically change him from Site 1 to Site 2 within Compliant Enterprise. Sites have plenty of valuable uses, but this potential issue should be considered as you think about policies based on them.

About Policy Sets

Once you have clearly isolated a business problem and formulated the virtual information barrier that would solve it, you need to design a set of policies that, taken together, creates the barrier you need, and solves the problem. Each of the five use cases represented by the predefined sample policies in Policy Author represents a policy set. As the use cases show, they can range quite a bit in their complexity. (For details, refer to the “Sample Use Cases” chapter of the *Policy Author 2.0 User’s Guide*.) In practice, this is the way you proceed with Compliant Enterprise: one problem at a time, solved by one set of policies.

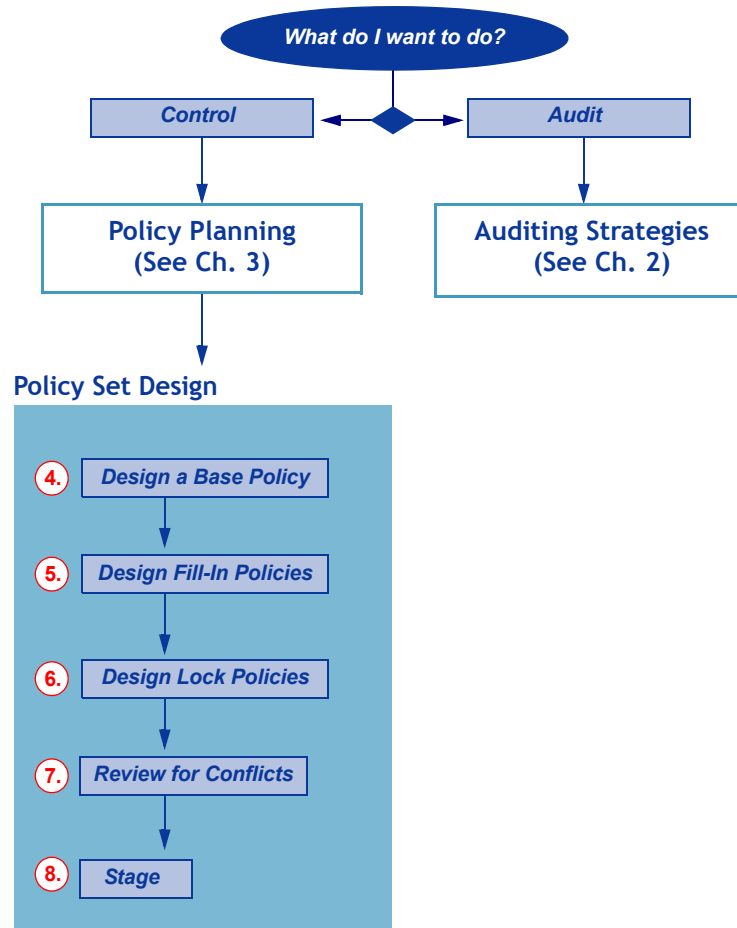


Figure 5-3: Policy Set Design

Defining a Base Policy

Each policy set should start with a *base policy*, which is broad enough to prevent a large part of the unauthorized use you are concerned about while still allowing access and use permissions to all appropriate users. This should be as comprehensive as you can make it; the closer your base policy can come to solving your problem, the better.

As a rule, base policies will be Allow Only policies, which define the only users who should have access to the specified resources. This may consist of one or more than one user component. The idea is to grant permission to a class of users first, then use additional Deny policies to block any outlier cases that should not be authorized, but for whatever reason cannot be excluded by the base policy. This is so because Compliant Enterprise's deny override design—that is, deny always trumps allow—does not let you deny access to a group of users and then grant access to some subset of it.

Because base policies should be Allow Only, you need to identify the a set of users that can be expressed as a user component, and that is broad enough to include everyone who needs access to the resources controlled by this policy. If it is a bit too broad—that is, includes outlier users who represent gaps in the policy—this can be corrected by adding other “fill in” policies to exclude them.

In defining a base policy, it is helpful to answer the following questions:

1. How can these resources be expressed as document components?

There are several broad approaches to defining a set of resources as document components. Most obviously, you can focus on the documents themselves. A somewhat mechanical document-based approach is to define categories directly based on file properties. Policies based on document properties have the considerable benefit of retaining control of documents no matter how much they are moved or duplicated. By default, the following properties are available for defining document components:

- Access Date (before/on or after)
- Created Date (before/on or after)
- Modified Date (before/on or after)
- Name
- Type
- Directory location
- Owner
- Owner User Component
- Owner LDAP Group
- size (=, !=, >, >=, <, <=)

In addition, you can define your own custom properties, which you can then also use in defining components.

A more flexible document-based approach focuses on functional categories of documents. These may be broad, file type-based categories—for example, text/word processing, graphic images, movies, spreadsheets, CAD drawings, etc.—or narrower, organizationally specific ones, such as quarterly sales forecasts, performance reviews, engineering design specifications, meeting minutes, and so on. This approach has the benefit of flexibility, but it does require you to find some way to identify individual documents as members of the component—file

names (using wildcard characters) or extensions, location, other properties such as owner or size, or some combination of these.

A different but very common approach is location-based: you can define document components based on network directory locations, so that any document in a specified location, whatever its file name or type or other properties, belongs to a certain class. This provides a highly dynamic situation, in which any document copied into a specified folder can instantly become subject to a policy, and comes under control. (This is the principle behind lock boxes and reverse lock boxes, which we discussed above—see [page 53](#).) Because you can specify many physical locations in the definition of a document component—including the universal MyDocuments directory, for local copies—this approach allows you to grant whatever degree of document mobility you require, while still maintaining strict control.

Removable Media

Compliant Enterprise provides a special, predefined component that allows you to represent any kind of data storage device that can be removed from a PC and transported to another location where its contents can be transferred. This covers any removable peripheral device that is visible as a drive in Windows Explorer; examples include removable hard drives, floppy drives, writable CD or DVD drives, tape drives, or USB ports. Note that removable devices that do not show as Windows drives but instead rely on proprietary applications to perform read, copy and paste operations, will not be included in this component.

2. Who are the users who need access to these resources?

It is important to identify carefully the people who do need access to each class of controlled resources. As we mentioned, you can define fill-in policies to exclude some users if the allowed user component is a bit too broadly defined, but you cannot grant access to excluded users if your base policy it is too narrowly defined.

3. How can I express these users as a user component?

As with documents, you can define user components based on one or more properties of the LDAP users imported into your Compliant Enterprise database. By default, these include:

- First Name
- Last Name
- Full Name
- User Principal Name
- Title
- Department Name
- Company
- Account Name
- Country Name
- ISO Country Code
- Numeric Country Code

You can also add custom user properties. You can combine more than one property using both AND or OR and IS or IS NOT operators, which provides great precision in defining the group of users you need. For example, you can define a component representing all senior engineers in the Research and Development departments your firm's branch offices in all countries except Korea and China:

All users, Title is Sr. Engineer OR Title is Senior Engineer AND Department is R&D OR Title is Research and Development AND Country Code is not Korea AND Country Code is not China

4. Which actions should be controlled?

Actions refer to the activity the policy is fundamentally designed to control: opening a file, copying it, e-mailing it, and so forth. Compliant Enterprise lets you define action components that represent one or more basic actions; [Table 5-3](#) describes all available basic actions. Note that actions are not customizable; you cannot add any actions to this list, though you can combine more than one in an action component. For example, you might create a component called *Transmit*, that combines the Send in E-mail, Send in IM, Copy/Embed, and Print actions.

Table 5-3: Available Basic Actions

| Action | Description |
|--------------------|---|
| Read | Open file |
| Rename | Change the name of a file |
| Write/Modify | Create a new file or change the contents of a file |
| Delete | Permanently remove a file from storage |
| Change Attributes | Modify file attributes, such as whether the file is read-only or hidden (using Windows file property dialogs) |
| Change Permissions | Change permissions granted to users of a file (using Windows security dialogs) |
| Paste | Copy and paste or cut and paste a portion of the file's contents to a location outside the file. Paste actions are enforced not at the point when a user selects the content and tries to copy or cut it. That is, the user is prevented not from pasting content per se, but from copying it in the first place. |
| Move | Delete a file from its current storage location and place it in a different location |
| Copy/Embed | Make a duplicate of a file/Insert a file into another file, such as by using the Insert File menu item to embed a spreadsheet in a word processor document |
| Attach to E-mail | Attach a file to an outgoing message in Microsoft Outlook |
| Attach to IM | Attach a file to an outgoing instant message in Yahoo Instant Messenger, AOL Instant Messenger (AIM), Microsoft MSN Messenger, or Microsoft Windows Messenger |
| Print | Print a file to a printer device or print to an output file |

5. Can these actions be controlled on the file server, or only on desktops?

This is an important point, since it underlies the distinction between access policies and usage policies. Because access policies are enforced on the file server only, they can include only the first six basic actions in [Table 5-3](#), above. Usage policies are broader, since they are enforced at the desktop level; they can include any basic action. Put another way, if your policy uses one or more of the first six actions, it can be enforced by the file server enforcer (an access policy); if it uses any of the last six actions, it must be enforced by desktop enforcers (a usage policy).

6. What other components are required for this policy?

Besides users, documents, and actions, there are two other available components, *applications* and *computers*. Both of these are optional for any given policy, and will not be used as frequently as the other three.

Application Components

Applications can be specified by name; for example you might define a component called *Authorized Spreadsheet Apps*, consisting of a list of applications approved for use with spreadsheet files; and then you can incorporate this component in a policy that allows only users using those applications to open spreadsheets.

Note that Compliant Enterprise provides two special, pre-defined application components:

- *Monitor Specific Applications*: Allows you to specifically list all the applications you want Compliant Enterprise to monitor, in connection with all deployed policies. This is useful if your implementation is limited or restricted enough that you can be confident which applications need to be monitored, and which do not.
- *Monitor All Applications - Exceptions*: Allows you to list specific applications that you do not want any policies to be enforced on. This is useful if PCs are running applications you know you will never want any policy to interfere with—a virus scanning application, for example.

Computer Components

Computers can be specified by one or more of the following default properties:

- DNS Host
- Machine Name
- Operating System
- Network Address
- Site

There is special predefined computer component, *All Computers with Agents*. This component represents all desktop and laptop PCs on which Desktop Enforc-

ers are installed. It can be very useful when you want to write a policy to ensure that only controlled computers are allowed to access sensitive documents—a commonly encountered case.

Because your organization may have enforcers running on a large number of desktops, the membership of this component is a potentially long list of computer names. For convenience, Policy Author's Action menu has a special command, *Update All Computers with Agents*, that lets you update the membership list all at once.

Users, Applications and Computers

There is one important point to understand about computer and application components: *they always work in combination with the user component as the singular grammatical subject of a policy.* That is, they are not independent components in the policy, but rather are always logically combined with the user, and must work together in the policy. (Note that this linkage is reflected in the way these three components are represented in Figure 5-1 on [page 47](#).)

For example, if you define a policy “Allow [HR Staff] using [Approved Spreadsheet Applications] on [PCs at the Global HQ Site]” to do some specified action, then all three of those components work together as the required subject of the policy. The policy will be enforced *only if all three* conditions are present. This means that the policy would *not* allow an HR staffer using an approved spreadsheet application on a PC at a branch office site, or an HR staffer on an unapproved application at the Global HQ site, to perform the specified action. So whenever you use application or computer components, be sure to consider the combination of all three as the subject of the policy, and recognize that they always work together.

7. What obligations are appropriate?

In the current release, Compliant Enterprise offers three kinds of obligations: log the event, display a custom message to the subject whenever the policy blocks the specified action; and send an e-mail notification to a specified administrator. For any given policy, you can include any combination of these four, or none of them. [Table 5-4](#) provides some guidance on which combination is useful in which cases.

Table 5-4: Notification Feature Settings

| Log Event | On Deny Notification | E-Mail Notification | Result / Recommended Use |
|-----------|----------------------|---------------------|---|
| X | | | Event logging is always enabled, regardless of which other notifications are used. Log Event only is the setting you would use for a “pure audit” case. |
| X | X | | This combination will remind subjects why their attempted action has been blocked, but will not send notifications to an administrator. Useful for reminding subjects about usage rules they might honestly have forgotten or overlooked, but do not have any serious or immediate sensitivity. |

Table 5-4: Notification Feature Settings (Continued)

| Log Event | On Deny Notification | E-Mail Notification | Result / Recommended Use |
|-----------|----------------------|---------------------|---|
| X | | X | Stealthier observation mode in which user is not notified of policy application, but an administrator is. Can be used when policy prevents access, or even when policy allows access but provides notification only. Especially useful in less benign cases, where some users may be engaged in more actively harmful activity, or even directly suspected of willful abuse of their information access permission. |
| X | X | X | For use in really serious cases, when administrators need to be notified of an attempted action that is so serious that it must be blocked in real time. |

Designing Fill-in Policies

By *fill-in policies*, we mean policies that work together with the base policy to cover exceptional or outlier cases of users or conditions that, for whatever reason, could not be covered by the base policy. These policies serve to fill gaps in the access or usage control—cracks in the virtual barrier, as it were. For this basic logical reason, fill-in policies will generally be Deny policies.

Allow Only vs. Deny Policies

Note that at first it may seem as though you can use several Allow Only policies working together in the same policy set. However, this will nearly always be a mistake, especially if their purpose overlaps. For example, you may want to restrict the use of sensitive HR documents to HR staff only, and in addition, to let the contract workers on the HR staff use these documents only on workdays. This might seem like two Allow Only policies:

P1: Allow only HR staff to access HR Docs

P2: Allow Only HR staff who are also contractors to access HR Docs between 8 a.m. Monday and 6 p.m. Friday.

However, in practice these will conflict with each other and produce undesirable results. Bear in mind that every Allow Only policy is also, logically, a deny policy: it will deny any user who is not consistent with its subject. Thus P2 will allow HR contractors to work as intended, but it will also block everyone else from accessing the documents, including the full-time HR staff who are the subject of P1—obviously an unintended effect. In this case, P1 is properly defined, and should serve as the base policy; the contractors should be handled by a Deny fill-in policy.

P1: Allow only HR staff to access HR Docs

P2: Deny HR staff who are also contractors to access HR Docs between 6 p.m. Friday and 8 a.m. Monday.

Design Questions

In defining fill-in policies it is helpful to pose the following questions:

1. What substantive gaps does the base policy leave uncovered?

Try to identify ways in which the base policy is not adequate by itself in solving the problem at hand. This generally focuses on exception cases, such as:

- Cases where you have to define the base Allow Only users a bit more broadly than required, for whatever reason. This is actually a very common case; as a rule you will have to let in some unauthorized users in order to cover all authorized ones. You then filter out the unauthorized ones with Deny fill-in policies.
- Issues of special contexts, such as time—whether there are times of the day, week or year when some exceptions to the base policy should be enforced.
- Anything special about the resources themselves that create exceptional cases that need to be covered by a fill-in policy—for example, approved drafts compared to working drafts, or documents whose sensitivity is related to external events such as press releases or product announcements.

2. Which additional users need to be blocked to fill in the gaps?

As with the base policy, you need to identify the users the policy is directed toward. You can define them based on properties, just as with base policy users. Often, these users may be a subset of the users specified by the base policy—those who should not be authorized under the policy set, but who need to be explicitly excluded because the base policy is somewhat too broadly defined. For example, you may need to allow all members of a certain project team full access and use of certain documents (your base policy), but then Deny the team members who work for an outsourcing partner firm from copying, printing or distributing the documents (a fill-in policy).

Don't forget that users may be combined with specified applications and computers to form the full subject of a policy. That is, when you are thinking about users you should also be considering whether your fill-in policy would benefit from including computer or application components.

3. What obligations are appropriate?

You can set obligations for fill-in policies just as you do for your base policy. The kinds of obligations you want will depend on the policies—they may be the same as the base policy, or they may differ.

4. What are the possible costs of this policy, and are they justified given the category of information we are dealing with?

As with any policy, you need to think carefully about the potential costs and be sure that they are appropriate to the sensitivity value of the information resources in question. Costs primarily consist of interference in the way people normally do their work: blocking access to documents they formerly could use, or preventing them from using documents in formerly permitted ways. Such changes should be anticipated—that is, they should be consistent with and con-

tribute to the explicit goals of the policy set—but nonetheless they can annoy users, and it will require time and resources to explain them, and to field inevitable requests for special access or usage permission.

Evaluating your Fill-in Policies

For every base policy, you will need to design several fill-in policies. However, you should be careful about the number and complexity of your fill-in policies. As we have mentioned, policy sets should rarely consist of more than half a dozen policies—preferably three or four. If it seems like you need to define a large number of fill-in policies, you might want to review the base policy—maybe a better designed base policy would leave fewer gaps to fill. Some ways you can improve your base policy include:

- Focus the policy more tightly by including more than one user component. Bear in mind that by combining many components with both AND and OR operators, and by using both IS IN and IS NOT IN to include components, you can design policies that cover intended groups of users more precisely.
- Consider whether the components themselves are appropriately designed. You may be able to improve the way a policy works by redefining one or more of its components, rather than the policy itself.

The category of information may be useful here, in identifying whether you have defined enough fill-in policies—that is, filled it in tightly enough. “Because this is Red information, we need to worry about every potential gap, no matter how unlikely.” If it were only Yellow, such comprehensiveness is less of a concern, especially when the costs of a policy are more significant.

Defining Lock Policies

By *lock policies*, we mean policies designed not to control the direct access or use of resources, but rather to block activities that might allow users to willfully circumvent access or usage policies. Lock policies focus on technical strategies such as renaming or copying parent directories, resetting file or folder permissions, transferring files after converting them to zipped archives, and so on. Because these activities by definition involve more active, conscious efforts to evade policy controls, they deserve careful consideration.

In defining lock policies, it is helpful to answer the following questions:

1. Are there any mechanical vulnerabilities in any policies in the set?

Mechanical vulnerabilities refer to anything that might allow users to actively circumvent the intent of the policy set and use information in unauthorized ways. Some examples to consider:

- moving or renaming parent folders or sites
- renaming files or portal content items
- copying parent folders to removable media

- changing file properties (especially if policies are based on file properties, such as create date or owner)

2. What actions do I need to control to correct those vulnerabilities?

This is closely related to the previous question, and the answer to it will flow directly from the nature of the vulnerabilities you discover. Some of the features that are particularly useful in lock policies include:

- The Rename, Change Attributes, and Change Security actions
- The Removable Media feature
- Directory locking (see below)

3. What obligations are appropriate?

Again, because the activities involved in evading policy controls are intrinsically more sneaky than the ones governed by simple access or usage policies, you will probably want to apply more stringent obligations (e-mail notifications) to these kinds of policies.

4. Are there any costs to these policies? Are they justified by the info type?

As with any policy, you should always be aware of any costs lock policies impose, and be sure they are justified by the seriousness of the resources being protected. You should also consider the effects of lock policies particularly carefully, to ensure that they do not accidentally prevent anyone from using resources in legitimate and permitted ways.

About Directory Locking

One important tool for designing lock policies is the Include Only Directories feature, which essentially allows you to define policies that apply to directories, rather than to the individual files in them. This is valuable in that it allows you to explicitly block users from moving, copying or renaming parent directories, which is a fairly obvious potential way of circumventing policies that are based on location. This property creates, in effect, directory components rather than document ones—they represent directories only, not files. When used in a policy, they will control actions taken on specified folders, but not the individual files within them.

This is a very useful feature because even if you have a policy that controls all files within a certain directory, the policy cannot stop a user from copying or moving a parent directory, thus circumventing the policy. This feature allows you to write policies that prevent this and similar problems.

For more details, refer to “Directory Components” ([page 55](#)).

Looking for Conflicts

The last step in refining a policy set is to carefully consider how any policies in this set conflict with one another, or with policies in any other set. One useful approach is to examine policies and sets from the perspective of components

common to them. That is, for a given user component, identify all policies that will control its access or use of resources, and consider whether the effects of these multiple controls will create any problems. One case to look for, for example, is if a user is the subject of an Allow Only policy governing some action, and also part of a Deny policy governing a related or overlapping action; such cases are likely to produce unexpected results.

You can then do the same for computer components, then documents, then applications: whenever a component is governed by more than one policy, make sure the policies do not impose incompatible or contradictory controls.

Guidelines for Design

Of course, the special circumstances of each organization's information control needs will play an important role in shaping the way Compliant Enterprise is used in designing and implementing policies. However, the following list presents a number of general rules that should be born in mind when designing components and policies, since they will produce more effective policies in nearly all circumstances.

Designing Policies

1. Less is more, and simpler is better.

As a rule, each policy set should comprise no more than four or five policies, and preferably two or three. Most information control problems should not take more than this; if they seem to require more, consider whether you have defined your problem—or your set of information resources—too broadly.

You should generally avoid the impulse to define long lists of very granular policies that address many specific cases. The more narrowly defined your policies are, and the more complex each policy set, the more likely they will be to conflict or interfere with one another, and the harder they will be to troubleshoot.

2. Start with the global, and move to more granular cases only as needed.

Give careful attention to defining a base policy that will prevent most of the users from prohibited information or uses, and only then focus on filling the gaps by defining more granular policies that will catch the exception cases the base policy misses. In an ideal scenario, you would start with a policy that prevents 80% of inappropriate access or use, then add a second that catches the next 15%, then a third that picks up the remaining 5%.

This point emphasizes the need for careful planning when designing policies. Remember that Deny policies take priority over Allow Only ones. This means that if a broad global policy inadvertently blocks a certain subset of users who should be authorized, no Allow Only policy can unblock them. Instead, you need to include exceptions in the broad policy (see below). This is why the base policy will nearly always be Allow Only.

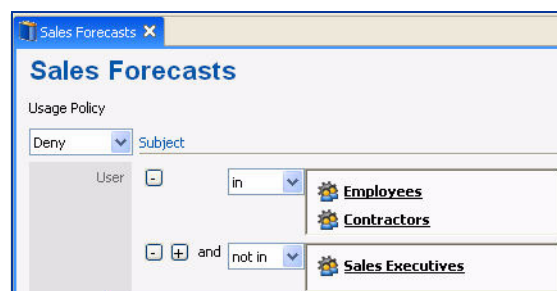
3. In Deny policies, always consider exceptions.

As we mentioned above, Deny policies cannot be overridden by Allow Only ones. This means that any Deny policies must be carefully designed to exclude as many inappropriate users as possible, but no appropriate ones. That is, exceptions to the deny policy must be included in the policy itself, since they cannot be handled by a separate policy.

It is easy to build exceptions into a single policy, by using the syntax

```
Deny Users In [NNN] Or [NNN] But Not In [NNN]
```

In Policy Author, this would look like the following:



4. Take advantage of User Groups, for a role-based approach.

Policies that are based on what users do rather than who they are will tend to be more flexible and reliable. One way you can do this, is by consistently basing policies not on properties of individual users, but by membership in groups. You are better off using the groups that are defined in Active Directory, since they can be automatically updated with a synchronize chron job; but you also have the option of defining custom groups directly in Policy Author. These will need to be manually changed and redeployed to reflect changes, but are still preferable to defining policies based on individual user properties.

5. User, Application, and Computer components work together as the policy's subject.

We mentioned this earlier, but it is important to reiterate that the subject of any policy consists of the specified user, computer, and application components combined. This means that, for example, a policy governing [contractors] using [approved accounting applications] on [corporate HQ computers] will be enforced only when all three of those conditions are met. You should not expect it to control contractors working on any branch office computers, or on some other applications.

Designing Components

1. Components should be as reusable as possible.

Do not define them to suit just one policy, but bear in mind their usefulness in multiple policies, both within and across policy sets.

2. Components should represent categories or classes, not individuals.

There is usually more than one way to define a component; when given a choice always choose properties that will be most flexible—meaning, properties that are less likely to need to be manually changed to reflect changes in your physical environment.

3. Favor components that can be automatically synchronized over those that cannot.

Underlying changes to users, user groups and hosts can be automatically propagated from the Active Directory to Compliant Enterprise, using a regular script that calls the Synchronize utility. This is an extremely useful feature, since it means that if a policy is properly designed, it will be automatically updated to reflect changes in your physical environment, without any requirement for manually editing and redeploying either components or policies.

(Bear in mind, this is only possible for users, user groups and hosts, since they are the only kind of components that are stored in Active Directory. Changes to Applications, Documents, Sites, and Actions must be manually updated and then redeployed to reflect changes. However, these kinds of components are by nature much more static than users, groups and hosts are.)

4. When defining components, favor dynamic properties over static ones.

This will make policies more flexible since they can be synchronized with AD. For example, Full Name is a static user property, while Department Name is a dynamic one.

5. Take advantage of custom properties.

Bear in mind that although users, groups and hosts are enrolled with a set of default AD properties, you can manually import whatever custom properties are defined for them in your Active Directory back end. The broader the selection of user, group and host properties you have available as you define user and host components, the more precisely you can define those components so that they cover exactly the information you want, without requiring you to define any exceptions.

Custom properties can be even more valuable in document components. You can manually define custom properties and associate them with documents, then define document components to represent all documents with some value for that property. This provides a very powerful tool for content-based policies. For example, you can associate a custom property, “CustomerInfo = Y”, with any documents containing customers’ personal information, and then write policies

that control who may use any documents in that class, and how. Especially when used in combination with third-party content analysis tools, custom document properties offer enormous possibilities for defining policies that are precisely focused yet flexible.

Policy Set Examples

This section looks at some example policies and shows common mistakes and how to fix them by following the policy guidelines described in the previous section.

1. Access & Use Control

Suppose you wanted to prevent a defined set of documents—approved employee performance reviews, let's say—from being deleted, modified, copied, or moved. You might define a policy as follows:

Prohibit all users from deleting, modifying, copying, or moving approved employee performance reviews.

However, this policy alone is not sufficient to follow the guidelines about preventing access from uncontrolled desktops and separating access and usage policies. As currently written, and acting on its own, this policy will have different effects depending on whether the user is using a desktop or laptop PC with a policy enforcer installed. If a user accesses an Approved Employee Performance Review from a PC that does not have a policy enforcer, or the policy is not deployed to that policy enforcer, he or she will be able to move or copy the document, since the computer system is uncontrolled.

To fix this deficiency, we have to separate the access and usage portions of this policy into three policies: one for access policy, one to prevent users on uncontrolled desktops from accessing the controlled documents, and one to control usage. This way, anyone who can get the documents will be under use control, and anyone not under use control will not be able to get the documents.

Deploy the first two policies to the Compliance Enforcer installed on the file server:

Deny all users delete or modify of Approved Employee Performance Reviews.

Deny Computers Without Agents read of Approved Employee Performance Reviews.

Deploy the third policy to the Desktop Enforcer installed on the PC:

Deny all users copy or move on Approved Employee Performance Reviews.

Note that this is a good example of how the stock component Computers Without Agents is used.

2. Duplication Control

To prevent users from creating copies that are not subject to policy control or moving a document to an uncontrolled location, create a usage policy and an access policy:

Deny all users duplication of Destiny Product Documents outside of Destiny Product Documents.

Deny Computers Without Agents read on Destiny Product Documents.

In this example, users will only be allowed to move or copy documents to locations that are defined as Destiny Product Documents. If a user attempts to copy or move a document outside of the specified set of locations, including posting it to a SharePoint site, the attempt will be denied.

3. Export Control

The following two policies illustrate the guidelines on using existing organizational terminology and preventing documents from being moved. Using the term “Source Code,” which represents how a software company views files of a certain type, these policies force all applications that create source code to do so within a given storage location, and prevent users from moving source code to any other location:

Deny Source Code Control Applications to create Source Code outside of Local Development Code.

Deny all users duplication or distribution of Local Development Code outside Local Development Code.

In this example, duplication is defined as paste, move, copy/embed, or rename, and distribution is defined as attaching to e-mail or instant message. Any export of files from the source code control application will be restricted to specified locations on desktop or server file systems. Once these files are exported, the second policy will ensure that they are not copied to uncontrolled locations or attached to an e-mail message.

Once you have carefully designed your policies and the components they will require, it is very simple to construct them in Policy Author, then deploy them throughout your enterprise with a few clicks. However, you should never fully deploy policies without first staging them in a controlled environment, where you can test them for design flaws, unexpected results, conflicts, and so on. In this chapter we will provide some advice on how to stage and test policies before you deploy them fully, as well as how to manage policies after full deployment.

The chapter is organized into the following sections:

- The Staging Environment ([page 73](#))
- Testing Policies ([page 75](#))
- Managing Policy Versions ([page 76](#))

The Staging Environment

There are two basic approaches to staging policies: using a limited deployment, or using a staging environment.

Limited Deployments

This option involves constructing components and policies on the production installation of Compliant Enterprise, but deploying them only to a strictly controlled number of hosts in your production environment. Based on how they work, you can make design changes in your policies, redeploy them, and continue testing the results.

This approach requires that you use care in identifying which policies and components are works in progress, still being tested on a small number of enforcers; and which are finished ones that have been tested, fine-tuned, and deployed to the whole production environment. For policies, you can do this with folders in Policy Author's policy tree structure—for example, create an In Progress folder where you can place all your unfinished policies. Because components are not organized in a folder structure, you have to identify unfinished ones by appending some indication—*test*, *beta*, etc.—in their name. You can then rename them once they are tested and you are ready to deploy them fully.

Staging Systems

The alternative staging option involves setting up a separate, isolated system where you can define policies and components, deploy them to test enforcers,

and monitor the results. The installation of Compliant Enterprise in the staging system can have access to your actual Active Directory and file servers, so all the objects in your policies will reflect the production network; however, all the enforcers in the staging system should be dedicated test machines, with hardware, platform and applications configured to represent production PCs as closely as possible. This arrangement allows you to construct, test, and refine policies without any danger of interfering with production processes or users. Once policies are fine-tuned, you can then create them on the production Control Center and deploy them to the production enforcers.

When you are transferring policies to the production system, even though they have been tested it is recommended that you control the deployment at first. You can do this in two ways:

- Deploy policies to a limited number of production enforcers, where you can monitor their behavior to confirm that there will be no problems. However, this will not be adequate for catching problems that may arise as a result of large-scale or high-volume operation.
- Deploy them to all production enforcers, but as When policies. This will prevent them from doing any potential harm, while allowing you to confirm that they are being enforced as designed. (You can use policy activity reports to do this—it amounts to a focused audit, as we described in Chapter 3.) Once you are confident the policies are working properly, you can change them to Allow Only or Deny, as appropriate, and redeploy them.

The Staging System approach is has the benefit of reducing risks to the production system, since no policy is deployed to a production server until it has been tested to your satisfaction. It does require more hardware resources and somewhat more effort than the Limited Deployment approach, and for this reason it may more suitable for larger organizations with more elaborate policy implementations.

Refining and Redeploying

The point of staging your policies is to test how they work. In most cases, this will involve discovering and fixing design flaws, or generally fine-tuning both policies and components through minor changes. Whenever you need to make any changes to deployed policies, you must redeploy the new version. The staging process will commonly involve repeated redeployments in this way.

Compliant Enterprise allows you to make changes to a deployed policy and then redeploy the new version, in effect overwriting the earlier version wherever it is deployed. However, a preferable approach during policy staging is to make a copy of any policy you want to change in some way, identify it by changing its name in a clear way (append “v2” to the name, for example), make your design changes in the copy, and then deploy the altered copy to a limited set of your test enforcers. This has the benefit of allowing you to limit the number of hosts the new version is distributed to, which limits the potential for interference if there is a problem with the changes in the new version. (If you simply redefine a policy and redeploy it, it will be distributed to all the enforcers where the earlier version is deployed.)

Testing Policies

Once you have deployed beta versions of components or policies, either to a limited set of enforcers in your live environment, or to your test enforcers in a staging system, you can start testing the way the policies work. This is pretty much a commonsense process, with two primary goals:

- Have users try to do things they are not authorized to do, according to some policy, and make sure the policy actually does block them; and
- Have users try to do things they are authorized to do, and make sure the policy does not interfere.

As a rule, rather than deploying and testing each policy in sequence, you should deploy all the policies of a policy set first, then run tests specifically on the activities each is intended to control. It is important to test each policy in the set while all the policies are simultaneously active; this is a good way to reveal possible conflicts, and to confirm that the policies work in combination with each other in the way the designers intended.

However, you should also spend time testing the general operation of PCs where policies are deployed, regarding all kinds of operations that are seemingly unrelated to the policies deployed there, using all the tools that users in the production environment would need to use. For example:

- Spend some time opening, working on, and closing files in all the applications installed on the test PC
- Test how well policies are enforced when large numbers of applications and documents are opened
- Log in as various users and as administrator, and try both permitted and prohibited actions
- Perform all kinds of non-file based activities while policies are deployed: copying, deleting and renaming directories; installing and uninstalling applications; sending e-mail and instant messages; resetting system settings such as the clock; changing folder security permissions; and so on.

The point here is to make sure your policies do not surprise you with any behaviors or effects that were entirely unanticipated. This is especially true in connection with any custom in-house applications you may be running, since by definition these cannot be tested with Compliant Enterprise before you install it.

Managing Policy Versions

Both when you are testing policies in a staging environment and after you have deployed them in your production network, it is often useful to keep track of different versions of a given policy or component. For example, if you notice that a deployed policy is not performing exactly as you expected, you can make some modification to it, deploy the new version, and monitor how it works. After that you may want to roll back to the original version, or make still further adjustments in a third version. To help with this, Compliant Enterprise has a sophisticated version tracking mechanism that automatically maintains all versions of any policy, deployed or not, and allows you to easily retrieve and redeploy earlier versions.

This feature is available through the Version History window in Policy Author, which you can open from the Action menu. For any policy or component you select, this window shows a list of all versions that have been defined; for each version, you can view the details of its definition and its current status, and a list of all desktop enforcers or file server enforcers where it is currently deployed.

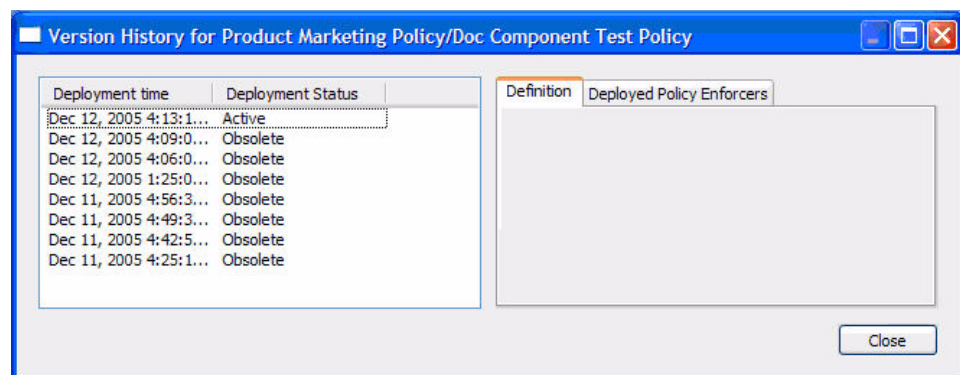


Figure 6-1: Monitoring Version History

Versions and Audits

One particular case where you will find version tracking feature useful, is if you are performing an information use audit and need to identify exactly which users had access to some class of documents over some specific period of time. In such cases, it is a simple matter to identify the version or versions of the policy that were deployed at the time in question (this is easy, since deployments are labeled by date), then examine the definition of the policy's user components for the period in question (component versions are also date-stamped). Whether you need to show that some users did have access or were actively blocked at some given time, the information is easily available no matter how many times the policy has been changed over time.

Monitoring Your System

The subject of this chapter, monitoring Compliant Enterprise, can be approached in two senses. On one hand, users need to be able to keep an eye on Compliant Enterprise's components, to make sure they are running and interacting properly at all times. These functions are performed with CE Administrator. In addition, users will need to monitor the performance and effectiveness of currently deployed policies, in order to know whether policies are properly designed or should be further fine-tuned. This is the function of the performance and system event monitoring tool, Reporter. You can use Reporter to monitor both general activity in the network (how users are accessing and using documents) and also policy enforcement activity (how your deployed policies are working).

The chapter is organized into the following sections:

- Monitoring Compliant Enterprise ([page 77](#))
- Monitoring Policy Enforcement ([page 79](#))
- Sample Reports ([page 83](#))

Monitoring Compliant Enterprise

Compliant Enterprise Administrator provides a continuous, real-time view of the status of all of the system's software components. It also provides a useful information about the status of policies: where they are deployed, and whether they are up to date. [Table 7-1](#) summarizes where you can perform these monitoring functions; for more details on using Administrator, refer to the *System Manual*.

Table 7-1: Monitoring the System

| How do I . . . | In Administrator: |
|---|---|
| Find out what hosts my Control Center server components are installed on? | Status tab, Status Overview link, Server Status pane (at right): Host column. |
| Check the health of all my Control Center server components? | Status tab, Status Overview link, Server Status pane at right: Last Heartbeat value turns red if expected interval has been exceeded. |
| Check how many enforcers are currently running? | Status tab, Status Overview link, Policy Enforcers statistics (middle left): shows File Server Enforcers, Desktop Enforcers, and Total. |
| Find out which enforcers are currently running? | Status tab, Policy Enforcer Status link: set Show filter to <i>All Policy Enforcers</i> . |
| Check if any enforcers are disconnected ? | Status tab, Status Overview link, Policies Statistics (lower left). |

Table 7-1: Monitoring the System (Continued)

| How do I . . . | In Administrator: |
|--|---|
| Find out which enforcers are disconnected ? | Status tab, Policy Enforcer Status link: set Show filter to <i>All Policy Enforcers with Warnings</i> ; then check the leftmost column for enforcers with Warning icons (exclamation points). |
| Find out which profile an enforcer is using? | Status tab, Policy Enforcer Status link: set Show filter to <i>All Policy Enforcers</i> , check Profile Name column. |
| Find out current heartbeat setting for an enforcer? | 1. Find which profile the enforcer is using (see above). 2. Policy Enforcer Configuration tab, Desktop Enforcers or File Server Enforcers link, locate the profile in the list at left, click the Settings tab, check the Heartbeat Frequency setting. |
| Check how many policies are deployed in my network? | Status tab, Status Overview link, Policies statistic (lower left). |
| Check whether any enforcers are using obsolete policies ? | Status tab, Status Overview link, Policy Consistency statistic (upper left). |
| Find out which enforcers are using obsolete policies ? | Status tab, Policy Enforcer Status link: set Show filter to <i>Policy Enforcers with Warnings</i> , look for red X icon in Policy Up-to-Date column. |
| Tell how often policies are being enforced? | Status tab, Status Overview link, Policies statistic (lower left). |

Monitoring Policy Enforcement

All the above material refers to monitoring the way the system is running, and whether policies are being properly deployed as you intend. The other sense of monitoring—measuring how well your policies are actually monitoring and controlling user activity in the network—is performed through a web application called Compliant Enterprise Reporter. Authorized users can use their web browser to open and use Reporter from any PC that can connect to the central installation of Compliant Enterprise.

About Reports

In Compliant Enterprise, a *report* is a formatted view of a particular set of information extracted from the Activity Journal—the internal data store where data on all system activity is maintained. You define a report by specifying the *query criteria*—that is, one or more filters for the data contents you want in the report—and the *runtime report settings*, which determine how the report will actually run. [Table 7-2](#) shows the available options for both these categories of settings.

Table 7-2: Report Definition Settings

| Setting | Description |
|--------------------------------|---|
| Query Criteria | |
| Search | Policy Activity = Report will show all qualified instances of policies being enforced, consistent with the other query criteria Document Activity = Report will show all activity in the system, consistent with the other query criteria—whether any policies were involved or not. |
| User | Specifies one or more users or groups whose activity this report will show. Click on the Search icon to open the search window, where you can select from all currently defined users and groups. |
| Action | Specifies one or more actions on which to filter this report. |
| Resource | Specifies one or more resources on which to filter this report. Click on the Search icon to open the search window, where you can select from all currently defined resources. |
| Policy | Specifies one or more policies on which to filter this report. Click on the Search icon to open the search window, where you can select from all currently defined policies. |
| Enforcements | Allows you to filter on enforcements: Allow, Deny, or both. |
| Runtime Report Settings | |
| Between | Defines the start and end dates of the time interval this report will cover. Click on the calendar icon to choose a date, or type it manually. |
| Show | Specifies the report display format: either an Event Details (tabular grid) or a bar chart grouped by policy, resource, time, or user. |

Report Types

Reporter can generate two basic types of reports:

- Document Activity Reports:** Provide information that is automatically recorded in the Activity Journal by Compliant Enterprise but is not necessarily related to policy enforcement. For example, you can generate a report of all cases when someone stopped a Desktop Enforcer, or read its log files; or when users logged in and out. The exact type of information available depends on the Document Activity Audit Level of the relevant enforcer profile(s).
- Policy Activity Reports:** Provide information that is directly related to enforcement of policies. For example, you could generate a report listing all users who tried to open documents that are restricted by a currently deployed policy.

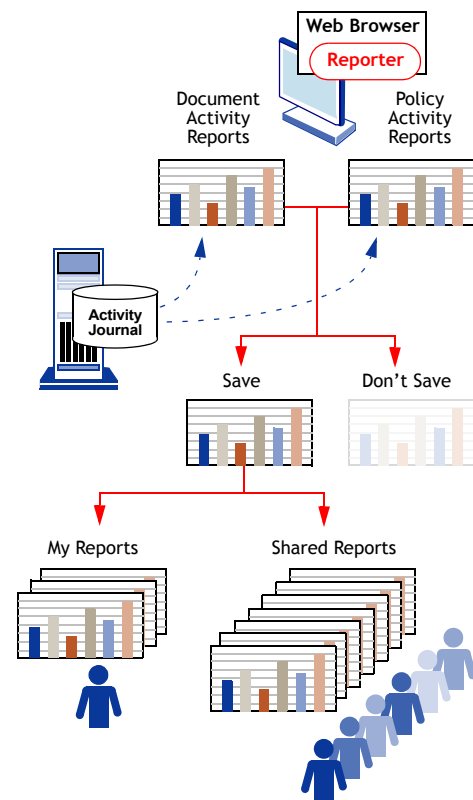


Figure 7-1: Reporting Functions

For either type of report, you have the option of creating it for a single viewing, or saving it so it can be reused later. If you save it, you have the additional option of saving it as your private, unshared report, or as a shared report, available for other authorized users to run later.

Action Filters

You can filter report output based on a wide selection of individual actions; for example, you can generate a report of all instances when any user copied files to removable media over some span of time—or tried to, but were blocked by an active policy.

The actions available for filtering will differ, depending on which kind of report you are generating. These differences are reflected in [Table 7-3](#) and [Table 7-4](#).

Table 7-3: Document Activity Reports: Action Filters

| Action | Report will include all instances of: |
|----------------------------|--|
| Enforcer Abnormal Shutdown | Any Policy Enforcer shutting down under any circumstances other than by the standard use of the shutdown executable. |
| Enforcer Shutdown | Any policy enforcer shutting down for any reason, normal or abnormal. |
| Enforcer Startup | Any Policy Enforcer starting up. |
| Change Attributes | Users changing attributes of any file covered by the policy. |

Table 7-3: Document Activity Reports: Action Filters (Continued)

| Action | Report will include all instances of: |
|-----------------------------|--|
| Change Security Settings | Users changing any of the security properties of any file covered by the policy. |
| Copy | Users copying any file covered by the policy. |
| Create/Edit | Users either editing an existing file covered by the policy, or creating a new file in any location covered by the policy. |
| Cut & Paste | All instances of users cutting or copying any content from of any file covered by the policy and pasting it into another document. |
| Delete | Users deleting any file covered by the policy. |
| Embed | Users embedding any file covered by the policy into some other file. |
| Move | Users moving any file covered by the policy from its current location to any other location. |
| Open | Users opening any file covered by the policy. |
| Print | Users printing any file covered by the policy. |
| Read Enforcer Binaries | Users opening any binary files associated with any policy enforcer in the system. |
| Read Enforcer Configuration | Users opening any configuration file of any policy enforcer in the system. |
| Read Enforcer Logs | Users opening any log file of any policy enforcer in the system. |
| Rename | Users assigning a new name to any file covered by the policy. |
| Send with E-Mail | Users attaching any file covered by the policy to an e-mail message. |
| Send with IM | Users attaching any file covered by the policy to an instant message. |
| User Login | Users logging in to any PC in the system. |
| User Logout | Users logging out of any PC in the system. |

Table 7-4: Policy Activity Reports: Action Filters

| Action | Report will include all instances of: |
|--------------------------|--|
| Change Attributes | Users changing attributes of any file covered by the specified policy. |
| Change Security Settings | Users changing any of the security properties of any file covered by the policy. |
| Copy | Users copying any file covered by the policy. |
| Create/Edit | Users either editing an existing file covered by the policy, or creating a new file in any location covered by the policy. |
| Cut & Paste | All instances of users cutting or copying any content from of any file covered by the policy and pasting it into another document. |
| Delete | Users deleting any file covered by the policy. |
| Embed | Users embedding any file covered by the policy into some other file. |
| Move | Users moving any file covered by the policy from its current location to any other location. |
| Open | Users opening any file covered by the policy. |
| Print | Users printing any file covered by the policy. |
| Rename | Users changing the name of any file covered by the policy. |
| Send with E-Mail | Users attaching any file covered by the policy to an e-mail message. |

Table 7-4: Policy Activity Reports: Action Filters (Continued)

| Action | Report will include all instances of: |
|--------------|---|
| Send with IM | Users attaching any file covered by the policy to an instant message. |

Using Reports

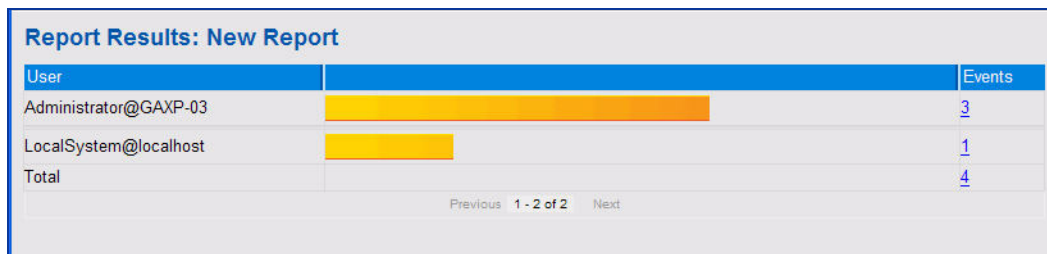
As the two tables above should suggest, you can derive great benefit from running periodical reports both on policy effectiveness, and on non-policy-related activity. You can create saved reports and run them periodically to monitor such events as:

- When a specific policy was enforced with a Deny, and on whom.
- When any of a set of policies was enforced with a Deny.
- When anyone tried to open a specific class of documents, but was blocked.
- When policies were enforced with an Allow, and on whom. This can be very useful in auditing projects, when you need to learn or demonstrate who had access to and actually used certain classes of documents.
- When specific actions were either blocked or allowed, in the context of a policy that covers multiple actions.
- When specific documents were handled in specific ways—opened, saved, copied, moved, etc.—and by whom.
- When the file property or security settings of specific documents were changed, and by whom.
- System events that are not associated with any policy—for example, abnormal shutdown of enforcers, normal shutdown or restart of enforcers, or users attempting to read logs or binary files of enforcers.

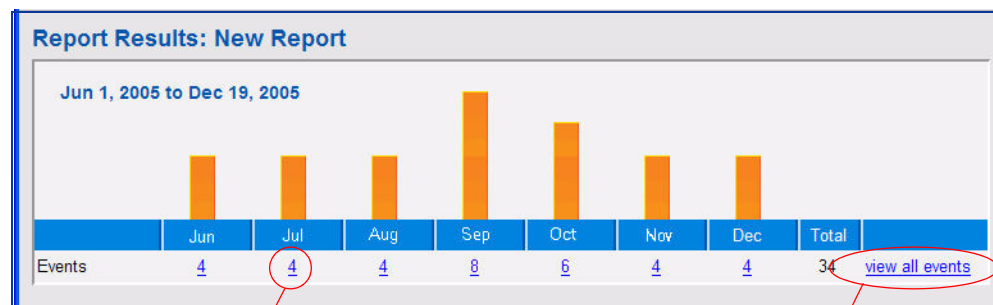
Sample Reports

The figures below provide examples of the two available report formats—grouped bar charts and detailed tabular data. Bar charts can be grouped by policy, resource, time, or user. Reports in any format are available in both onscreen and printer-friendly versions.

Bar Chart, Grouped by User



Bar Chart, Grouped by Time



Click on Total
to drill down
to detailed data

Click for detailed data
for whole report

Tabular Detail Grid

Click on any heading
to sort by that field

| Date | From Resource | To Resource | Action | User | Host |
|---------------------------|--|-------------|--------------------------|-----------------------|----------------|
| Dec 19, 2005 - 4:41:07 PM | agentprofile.xml /local/host/c:/program files/compliant enterpri... | | Read Agent Configuration | Administrator@GAXP-03 | gaxp-03.test.b |
| Dec 19, 2005 - 4:41:06 PM | commprofile.xml /local/host/c:/program files/compliant enterpri... | | Read Agent Configuration | Administrator@GAXP-03 | gaxp-03.test.b |
| Dec 19, 2005 - 4:40:52 PM | Compliance Agent Service.EXE /local/host/c: | | Agent Startup | LocalSystem@localhost | gaxp-03.test.b |
| Dec 19, 2005 - 4:39:26 PM | Compliance Agent Service.EXE /local/host/c: | | Agent Shutdown | Administrator@GAXP-03 | gaxp-03.test.b |

Figure 7-2: Sample Report Formats

In this appendix we will describe best practices for handling ongoing troubleshooting tasks. This topic can be usefully approached as a sequence of questions you ask and answer, in order to narrow down possible sources of the problem, to eventually discover and correct it. We refer to this as the standard debugging procedure.

The Standard Debugging Procedure

As you start creating and testing policies, you will likely encounter cases when you define a policy that seems to be structurally correct, yet when you deploy it to your test hosts, you do not see the expected document control results. Your policy may not stop a user from doing some action you intended to prohibit, for example, or it may block some action that should be allowed. In such cases, you can perform the following steps to isolate, understand and correct the problem. We'll refer to this as the standard debugging procedure for Compliant Enterprise policies. (There is also an HTML-based troubleshooting tool that leads you through this procedure; it is available on your Compliant Enterprise installation CD.)

Where to Start

There are two broad classes of potential problems with policy behavior:

- Technical problems that prevent your policy from being deployed properly; or
- Design problems in the policy that make it work differently from what you expect.

In the debugging procedure, you should focus on one or the other of these approaches depending on the results you see when you test your policy. If you do not see any behavior on the desktop at all, that could be caused by either type of problem. In this case, start by eliminating the possibility of technical problems first, then moving to possible for design issues. If, on the other hand, you see some behavior when you trigger the policy, but it is not the behavior you expected, at least you know that a policy is deployed and is being triggered in your test. Accordingly, you can conclude that your problem is with the design of the policy or its components, rather than any technical issues.

This basic dichotomy is represented in the first question, at the upper left in [Figure A-1](#). This figure represents the steps in the standard debugging proce-

Figure A-1: The Standard Debugging Procedure. All the numbered steps summarized in the figure are described in more detail below.

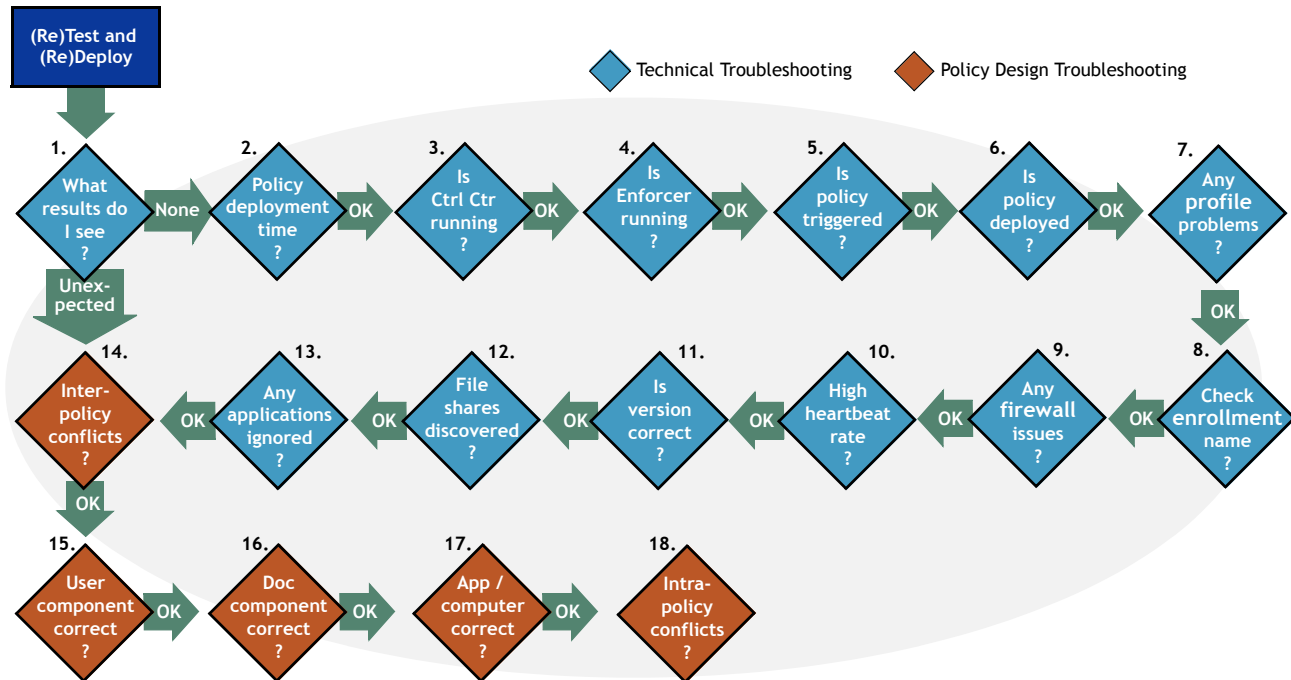


Figure A-1: The Standard Debugging Procedure

Identifying Technical Problems

Your first step is to deploy the policy and then perform some action on the test PC that you expect will trigger the policy. If you do this and see no results at all, the problem might stem from some technical problem in the system, or it may lie with a design flaw in the policy itself. Begin troubleshooting by eliminating the possibility of technical problems that might be interfering with the operation of your Compliant Enterprise system, or with the proper deployment of the policy you are testing.

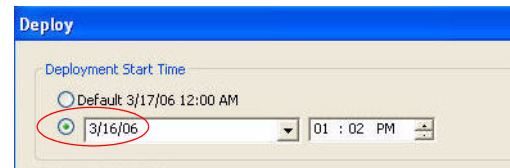
The numbers in the steps below refer to the sequence of questions in the diagram.

1. What results do I see?

When you test the policy, you will see one of two basic possibilities on the test PC: either the policy will not produce any effect at all, when one is expected; or it will produce some effect, but not the one you expect the policy to produce. In the first case, you need to eliminate the possibility of a technical problem that is preventing the policy from being deployed or working properly. In the second, you know that a policy is deployed, so you can focus on possible design flaws that affect the way it works.

2. Deployment start time?

If you do not see any results of your policy, it may be because it was not actually deployed yet. One common, easily fixed possible cause is the failure to set the Deployment Start Time to the present time. This is controlled by the radio buttons at the top of the deployment screen, as shown. The default setting is the top button, which is midnight of the current day; if you don't change that, you may think you have deployed your policy when in fact it is still in Pending Deployment status.



To check this, open the policy in Policy Author and redeploy it, making sure you click the bottom radio button as shown. The time for this setting is editable; by default it is the current time.

3. Is the Control Center running?

If you did set to deploy Now, the next step is to make sure all Control Center software components are running normally. To do this, open Administrator and check the Status tab, Status Overview link. The right-hand side of this screen displays current status of all components. Compare the Last Heartbeat timestamp with the Expected Heartbeat; if it is past due, it means the component has stopped running and needs to be restarted. In this case, simply restart the whole Control Center. Then wait long enough for the heartbeat interval to expire, and retest the policy.

StatusUsers And RolesPolicy Enforcer Configuration

Status OverviewPolicy Enforcer Status

Status Overview

System Status

Last updated: 7:56 PM

Policy Enforcer Status

(Last 24 Hours)

Policy enforcers not connecting0

Policy Consistency

Policy enforcers with out-of-date policy0

System Statistics

Last updated: 7:56 PM

Policy Enforcers

File Server Enforcers Registered1

Registered

Desktop Enforcers Registered3

Server Status

| Server | Type | Host | Port | Last Heartbeat |
|------------------|---------------------|---------|------|---------------------------|
| GATEST3_dms | Management Server | GATEST3 | 8443 | Dec 10, 2005 - 7:56:08 PM |
| GATEST3_dabs | ICENet Server | GATEST3 | 8443 | Dec 10, 2005 - 7:56:49 PM |
| GATEST3_mgmt | Administrator | GATEST3 | 443 | Dec 10, 2005 - 7:56:47 PM |
| GATEST3_reporter | Reporter | GATEST3 | 443 | Dec 10, 2005 - 7:56:08 PM |
| GATEST3_dps | Policy Server | GATEST3 | 8443 | Dec 10, 2005 - 7:56:44 PM |
| GATEST3_dac | Intelligence Server | GATEST3 | 8443 | Dec 10, 2005 - 7:56:45 PM |

Figure A-2: Checking Status of Control Center Components

4. Is the enforcer running?

If all Control Center components are normal, the next step is to confirm that the enforcer you are testing on is running. If you are testing an access policy, this will be a file server enforcer; if you are testing a usage policy, it will be a Windows Desktop Enforcer. In the former case, you will need to know which file server host your policy is supposed to be enforced on.

- For File Server Enforcers, open Administrator, go to the Status tab, Policy Enforcer Status link, and either search by host name, or filter by All File Server Enforcers. Check the first column in the grid to view the current status of the file server enforcer you are interested in; an exclamation point icon indicates a problem. (You can also check desktop enforcers from this tab.) If it is not running, you can restart it locally, through the standard Windows services manager in the Control Panel, or by rebooting the server if that is practical. Wait long enough for the heartbeat interval to expire, then retest the policy.

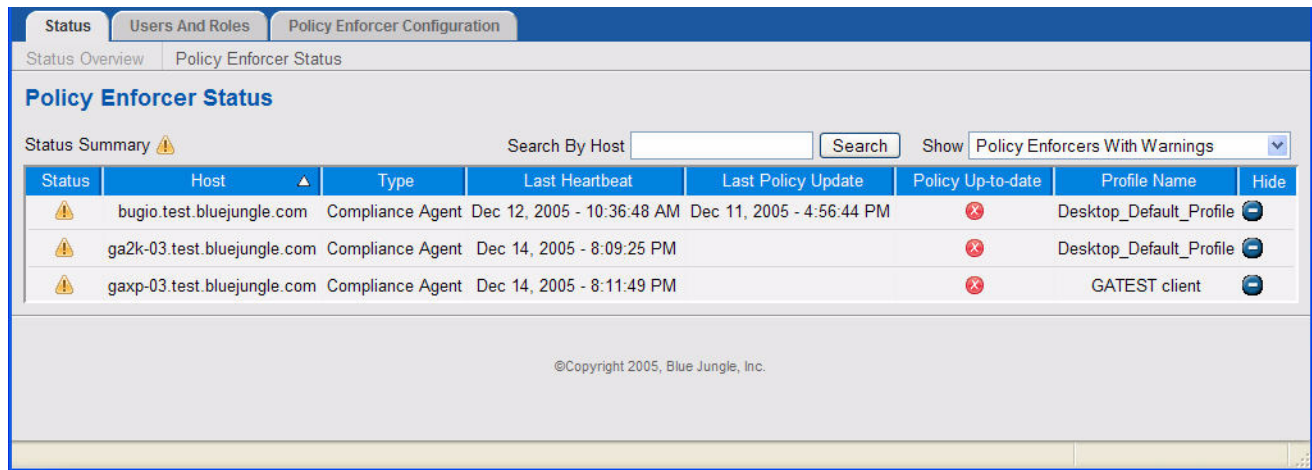
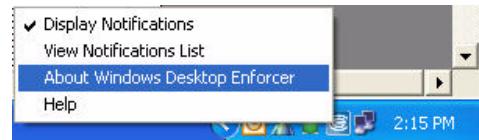


Figure A-3: Checking Status of Policy Enforcers

- For Desktop Enforcers, you can right-click the CE icon in the system tray on the local host, then select About Windows Desktop Enforcer. This will open a window with an *Agent Status* display: Running, or Not Running. If Not Running, restart the enforcer by rebooting the PC. Wait long enough for the heartbeat interval to expire, then retest the policy.



5. Is the policy being triggered?

Once you know the enforcer is running, you should determine for sure whether the policy is being enforced but is unexpectedly returning an Allow result—and therefore is not producing any noticeable effect—or is not being enforced at all. To do this,

- Change the policy's On Allow obligation to Log. This will create a record of every time the policy is enforced, even if the enforcement does not deny any action.
- After testing the policy a few times, use Reporter to run a policy activity report, filtering for just the policy you are troubleshooting. This will show whether the policy is being enforced or not. If it is, but is just giving you an unexpected behavior, most probably it is a policy design flaw, and you can skip to step NN. If it is not being enforced, you need to check whether the policy is deployed at all on that enforcer.

6. Is the policy deployed?

If the policy is not being triggered, it may be because it is not deployed at all on that enforcer. To confirm that the policy is deployed:

- For policies on either file server or desktop enforcers, you can open Policy Author, select the policy, then click the Deployment Status command from the Actions menu. This window has two tabs, listing all File Server enforcers and Desktop Enforcers. Select an enforcer in the Host list, to display all the components and policies currently deployed on it, in the Contents list.

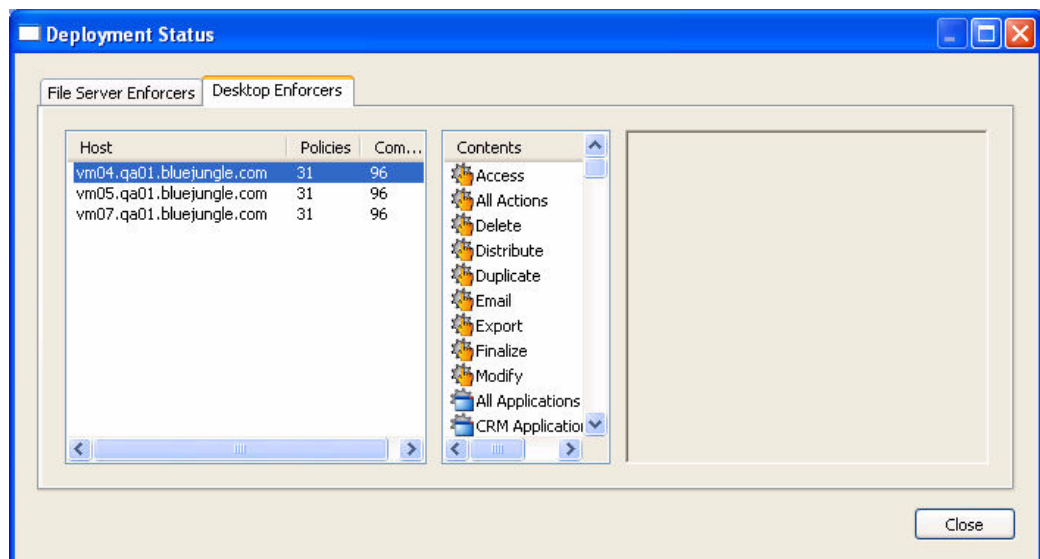


Figure A-4: Checking Deployment Status

- For desktop enforcers, you can also right-click on the CE icon in the system tray on the local host, then select About Windows Desktop Enforcer. This will open a window with a Policies Deployed On display, which shows the date and time the most recent policy was deployed to this enforcer. If this time is after you deployed the policy, you can be confident that your policy was properly deployed to this enforcer.



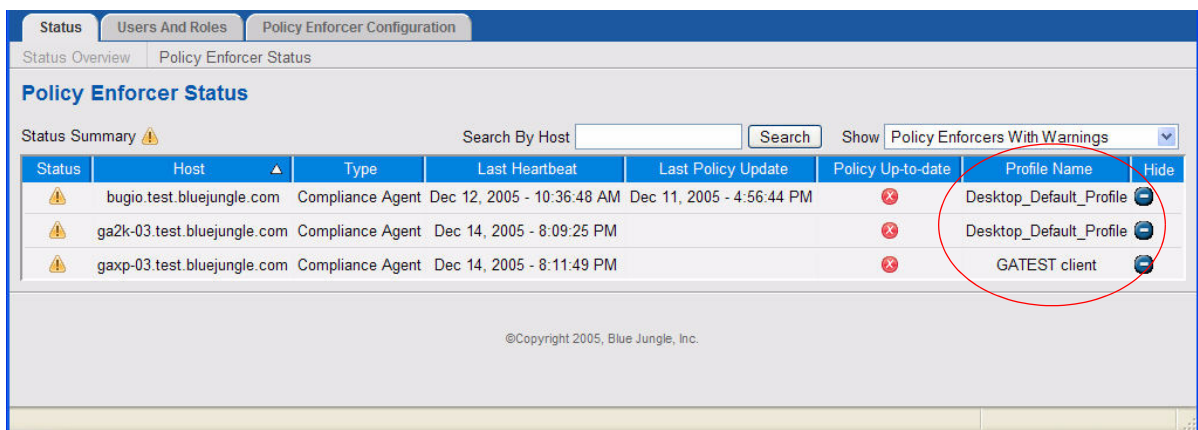
If you confirm that the policy is properly deployed to this enforcer, you can proceed to Is the version correct? ([page 94](#)).

7. Are there any enforcer profile problems?

If you determine that the policy has not been deployed to your enforcer, one possible cause is an enforcer profile problem. ICENet servers are the gateways for distributing policies to all enforcers, and each enforcer must be configured to the proper ICENet server—one that is part of the Control Center where you are defining policies for that enforcer. If it is connected to the wrong ICENet Server, the Control Center cannot distribute policies to it.

This connection is controlled by the enforcer profile associated with your enforcer host. Troubleshooting this involves finding out what profile your enforcer is using, confirming that the profile really is assigned to the enforcer, and then checking that the profile is assigned the correct ICENet server. To do this,

1. In Administrator, click the Status tab, then the Policy Enforcer Status link.
2. Set the Show filter to All Policy Enforcers, and check the Profile Name column for the enforcer you are troubleshooting.



| Status | Host | Type | Last Heartbeat | Last Policy Update | Policy Up-to-date | Profile Name | Hide |
|--------|-----------------------------|------------------|----------------------------|---------------------------|-------------------|-------------------------|------|
| ⚠ | bugio.test.bluejungle.com | Compliance Agent | Dec 12, 2005 - 10:36:48 AM | Dec 11, 2005 - 4:56:44 PM | ✗ | Desktop_Default_Profile | 🔍 |
| ⚠ | ga2k-03.test.bluejungle.com | Compliance Agent | Dec 14, 2005 - 8:09:25 PM | | ✗ | Desktop_Default_Profile | 🔍 |
| ⚠ | gaxp-03.test.bluejungle.com | Compliance Agent | Dec 14, 2005 - 8:11:49 PM | | ✗ | GATEST client | 🔍 |

Figure A-5: Checking Profile Name

3. Once you know which profile your enforcer is using, click to the Policy Enforcer Configuration tab, and select that profile from the combo-box at the left of the window.
4. The profile screen's ICENet Server field displays the host of the ICENet Server this enforcer is configured to. If this is correct, proceed to the next step. If it is not, review the profiles to find which one does use that

ICENet server, and assign the enforcer host to that profile. (You do this on the profile's Hosts tab.)

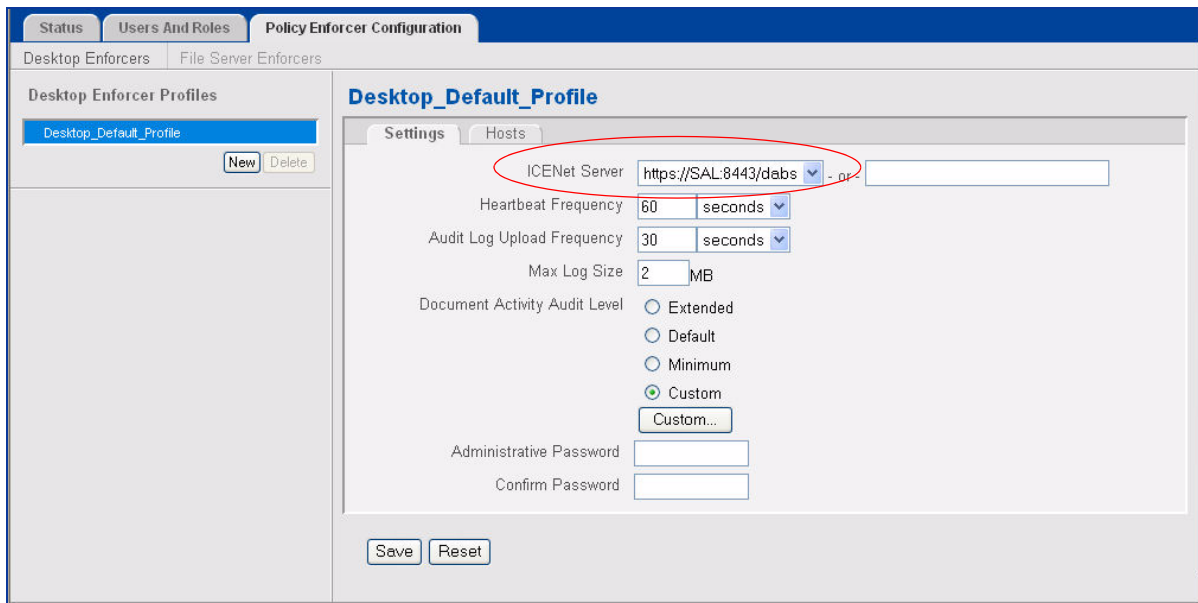


Figure A-6: ICENet Server for Each Profile

8. Is the enrollment name correct?

When you enroll the LDAP directory entities into Compliant Enterprise, you must supply a name for the enrollment—this is the *-n* argument for the Enrollment Manager's *Enroll* command. The value supplied here must be identical to the domain name as defined in the DNS.

If you make a typo in this domain name parameter, the enrollment will complete successfully because the Enrollment Manager cannot distinguish between a mistyped domain name and a correct one, but the enrollment will not map to the correct domain. As a result, Compliant Enterprise will not be able to deploy any policies to the enforcers in the domain.

To check whether this is the source of your problem, use the Enrollment Manager's *list* command to display a list of all current enrollments. To do this, open a command line window, change to the directory where Enrollment Manager is installed (by default, Program Files\Compliant Enterprise\tools\enrollments), and type the command

```
enrollMgr list
```

This will return a list of all current enrollments. Locate the one you are testing, and carefully check it against the actual name of the domain in which you are deploying policies. If the enrolled name is not correct, perform another enrollment using the correct name, then redeploy and retest your policies.

For details on using the Enrollment Manager, refer to Chapter 7 of your Compliant Enterprise 1.5 System Manual.

9. Are firewalls interfering?

Firewalls can be one reason a policy doesn't deploy properly. On every host where a Control Center server component is installed, you should be aware whether a Windows firewall is running or not. If it is, you need to configure the firewall to allow connections to the specific ports the Control Center uses. This involves defining three firewall exception ports in the Local Area Connection for each host. Note that these connections are unidirectional—they need to be opened in the server direction only.

The ports you need to allow access to include:

- Web Services port: 8443
- Web Applications port: 443
- Auto-discovery port: 19888

Once you define these exceptions, try redeploying the policy and then retest.

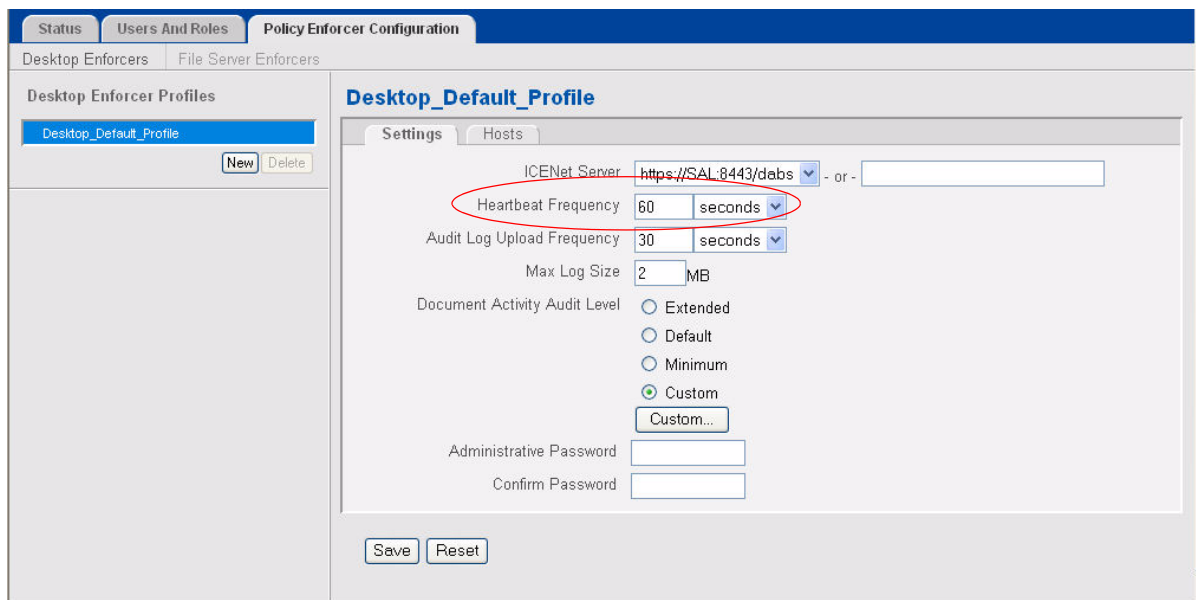


Figure A-7: Checking Heartbeat Frequency

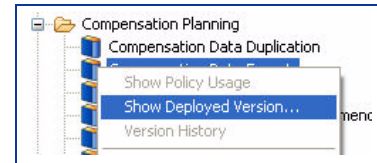
10. High heartbeat rate?

If the heartbeat frequency for this policy was reset to a very high value, newly deployed policies will experience a long delay before actually being distributed to enforcers where they take effect. If your policy has not been deployed, this is a possible reason. The default rate is 60 minutes; to check the heartbeat rate of the enforcer where you are testing, open Administrator, go to the Policy

Enforcer Configuration tab, and select the profile associated with your test enforcer. The Heartbeat Frequency for that enforcer will display near the top of the Settings tab.

11. Is the version correct?

If the policy is deployed properly and is being triggered, there may be a version problem. If you have made changes to the design of a policy, it may be that an earlier version is active but the changes in your more recent version have not been deployed. To check this, right-click the policy in Policy Author and select Show Deployed Version. If the details displayed here suggest an earlier version, redeploy the current one, then retest.



If you know there is only one version of this policy (for example, if it is newly created), you can skip this step.

12. File shares discovered?

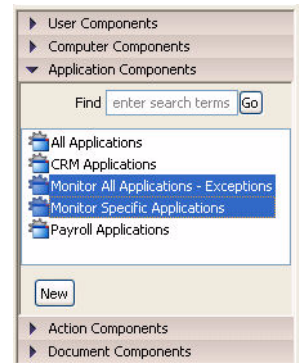
Next, check for undiscovered file shares. If you have not run the file share enrollment tool, *ResourcePathDiscovery.bat*, on the host where the resources you are working with are stored, there may be multiple file shares that allow users to access resources even if they are properly blocked by your policy. This utility must be run on every PC that you want to check for file shares—that is, on every host where resources you want to protect are stored. (Typically, this means network file servers.) File shares may be the source of your problem if:

- the Resource Path Discovery process never was performed in your system;
- it was run, but the file server you are working with was not included in the machine list file at the time; or
- your system's file share architecture has changed since the time the utility was run.

If you are confident that none of these is the case, you can skip this step. If you are not sure, re-run the Resource Path Discovery utility, making sure the file server you are working with is included in the machine list file, then redeploy and retest your policy.

13. Any application excluded?

Compliant Enterprise's Monitor All Applications—Exceptions feature allows you to globally ignore specified applications from enforcement by any policy. This feature works via an application component. Your policy may involve an application that has been excluded from enforcement by some previously defined component you do not know about. Open Policy Author, and use the Find Component search tool to look for any deployed component containing the strings **monitor** or **exception**; this should quickly identify any components that define excluded applications.



Identifying Design Flaws

If you checked all the possibilities described above and did not find any problems, it is likely that there is a design problem in the policy itself that is preventing it from working properly. At this point, you need to examine the policy's contents.

This is where you will start your troubleshooting in cases when you are seeing some policy enforcement behavior, but it is not the behavior you expect. In that case, you can rule out the possibility of any of the technical problems we discussed above.

14. Inter-policy conflicts?

Next, check for conflicts among several policies that may be deployed. That is, make sure the results you are seeing are caused by the policy you are testing, and not some other one. It is often possible that another deployed policy is triggered at the same time as the one you are testing, and is actually producing the enforcement results you think are coming from your policy. For example, you may be testing an Allow Only policy, and not know that there is a Deny policy already deployed that covers the same user(s) or resources. In such a case, the deny effect will take priority; but when you see the behavior, you may (quite understandably) mistake it for your Allow Only policy not working properly.

To troubleshoot this possibility, turn on the Notification Message obligation in your test policy, add the policy name to the notification message text, redeploy, and retest. This way you can always be sure what policy is being enforced on the desktop. (This is a good practice to follow in all policies.) If it turns out that some other policy is being triggered, carefully examine the details of that policy until you understand the nature of the conflict.

You can also check for inter-policy conflicts in Policy Author, by right-clicking the User component for your test policy, then selecting Show Policy Usage. This will display a list of all policies that use the component; by examining their details carefully, you may identify conflicts. You can do the same for the resource(s) in your test policy.

15. User component correct?

If you determine that it is the correct test policy that is being triggered, there may be a problem with a definition of one or more components in the policy. Start with the User, and check to confirm that the component as defined really does cover the user logged in on the test PC. To do this, open Policy Author, select the component, and select the Preview command in the Window menu. If the test user does not appear in the list of users in the Preview pane, there is a flaw in the definition.

16. Document component correct?

If the user component looks good, repeat the Preview procedure for the policy's document component.

17. App or computer component?

If the document component looks good, repeat the Preview procedure for the policy's application and/or computer components, if any. (These components are optional, and your policy may not contain any.)

18. Intra-policy conflict?

It is possible to specify multiple subject and resource components in a policy, using both AND / OR and IN / NOT IN operators. This allows you to create very complex definitions, such as "All users in X, Y, and Z but not in A or B." The more complex a subject or a resource is, the greater the possibility of a conflict within the definition of the policy. You cannot use the Preview feature to test combinations like this—it only works on one component at a time. Instead, try breaking them up and test them individually, at least ensure that simple policies are enforced. Also, pay extra close attention to this kind of compound subject or resource, and carefully consider whether the components may conflict with one another in an unexpected way.

One specific possibility to watch out for is specifying more than one path-based document component as the resource, but linking them with an AND operator. For example,

```
All spreadsheets in MyDocuments\Forecasts\**  
AND in MyDocuments\Forecasts\2007\**
```

Since documents normally are not saved in two locations simultaneously, this condition will never be met, and the policy will never be enforced. To correct it, change the operator to either OR or BUT NOT IN, or redefining the resource altogether.



Index

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

A

- action components 49
- All Computers with Agents 62
- application components
 - as part of subject 63
- applications
 - excluding from monitoring 62
 - monitoring 62
- attach to e-mail (basic action) 49, 61
- attach to IM (basic action) 49, 61
- Audit/Alert policies 48, 51
- auditing policies 25
- auditing strategies 28
 - focused audits 25
 - global audit 20
 - ongoing 30
 - security audits 27
 - summary of 18
 - targeted audits 22
- audits
 - using policy version tracking 76
- autosynchronization 56

B

- barrier design questions 35, 41
- base policies
 - defining 58
- basic actions 49
 - details about 61
 - list of 49

C

- canned components
 - ignored application 62
- change attributes (basic action) 49, 61
- change permissions (basic action) 49, 61
- components
 - default properties 50
 - design principles 70
 - ignored application 62
 - types of 49
- components and policies
 - overview of 45
- computer components
 - as part of subject 63
 - default properties 62
- control categories
 - color-code model 38
 - overview 37
- copy (basic action) 61
- copy/embed (basic action) 49
- costs
 - of fill-in policies 65
 - of information control 42
- creating
 - reports 80
- custom obligations 48, 53
- custom properties 70

D

- databases
 - supported 12
- delete (basic action) 49, 61

deployments, limited 73

design principles
for components 70
for policies 68

Desktop Enforcers
installation requirements 12
silent mode 24, 54

directory components
creating 55
example of 55

directory locking 67

document activity
audit level 20

Document Activity reports 26

document activity reports 80

document components
default properties of 59

E

effects
choosing 51

e-mail 61
see also Send in E-Mail

embed (basic action) 61

enrollment
synchronizing 56

exceptions, in policies 69

F

File Server Enforcers
system requirements 12

fill-in policies 59
designing 64
evaluating 66

filters
in report queries 26

focused audits 25
reports on 26

G

global audits 20
limitations of 21
reports on 20

goals (of policies) 46

I

Ignored Application 62

IM
see Send in IM

implementation
overview of 13

Include Only Directories
description of 67
using 55

information categories
control levels for 40
green 38
red 38
yellow 38

information control
costs of 42

information control requirements 36

Information Resource Model
synching with AD 56

L

limited deployments 73

lock policies
defining 66

lock-boxes, using 53

logical operators 51

M

Modify (basic action) 61

monitoring specific applications 62

move (basic action) 49, 61

N

nested properties 52

notification mode 54

O

object components
types of 49

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

obligations
 custom 48
 for fill-in policies 65
 in base policies 63

ongoing audits 30

operators
 using 51

P

paste (basic action) 49, 61

peripheral devices, controlling 60

policies
 allow only vs. deny 64
 base, defining 58
 design principles 68
 effects of
 choosing 51
 example purpose 46
 exceptions 69
 fill-in, designing 64
 lock policies 66
 managing versions of 76
 role-based 69
 staging
 approaches to 73
 structure of 46
 examples 50
 subject of 63
 testing 75
 troubleshooting 71
 typical goals of 46

policies and components
 design sequence of 48
 overview of 45

policy activity
 reports 80

Policy Activity reports 26

policy planning process, summary 31

policy sets
 designing 58
 examples of 71

Print (basic action) 49

print (basic action) 61

properties
 custom 70
 nested 52
 of components

static vs. dynamic 70

Q

queries 79
 overview 79

questions
 about virtual barrier design 35
 for designing base policies 59
 for virtual barrier design 41
 in designing fill-in policies 64
 in designing lock policies 66

R

read (basic action) 49, 61

releasing information
 definition of 39

removable media 60

rename (basic action) 49, 61

reports
 creating 80
 definition of 79
 sample queries 21
 types of 80

requirements, all hardware 12

reverse lock-boxes
 using 53

role-based design 69

S

security audits 27
 actions focused on 26

silent mode
 suggestions for using 24, 54

sites
 autosync issues 57

sizing requirements 12

staging policies
 approaches to 73

staging systems 73

summary table 28

synchronizing enrollments
 automatically 56

Index

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

T

- targeted audits 22, 23
 - limitations of 23
- time factor, in policies 39

U

- user components
 - default properties 60
- users
 - spectrum of 17

V

- versions
 - of policies 76
- virtual information barriers
 - designing 35
 - explanation of 33

W

- wildcard characters 52
- write (basic action) 61
- write/modify (basic action) 49