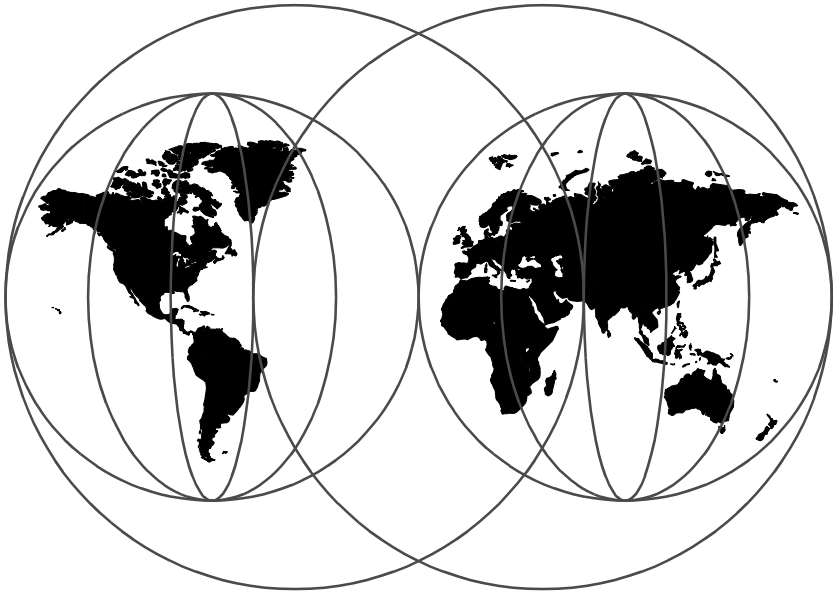
Tivoli User Administration Design Guide

Richard Hawes, Esau Moises Acevedo



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5108-00

SG24-5108-00

International Technical Support Organization

Tivoli User Administration Design Guide

July 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, “Special Notices” on page 115.

First Edition (July 1998)

This edition applies to the Tivoli program product, "Tivoli User Administration". The contents are not intended to be version specific (unless otherwise noted). It has been written with versions up to and including 3.6 in mind.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. DHHB Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

See also the preface section “Comments Welcome” on page xii about other ways to submit comments.

Contents

Figures

Tables

Preface

The first redbook in this series, *Getting Started With TME 10 User Administration*, SG24-2015, described the product and was aimed at helping the reader become familiar with the product features. However, an effective implementation of Tivoli User Administration involves planning and the design of a user administration policy.

This publication provides a methodology for designing Tivoli User Administration installations. Starting with a general description of what is required, we show how to define an administration hierarchy and how to implement effective user management.

We have aimed to talk about general methodology topics applicable to all platforms rather than discuss details of implementation on each platform (Windows NT, UNIX, OS/390 Security Server, and so on). Where we do address platform-specifics, this edition tends to discuss Windows NT and UNIX.

The Team That Wrote This Redbook

This redbook was produced by specialists working at the International Technical Support Organization (ITSO), Austin Center.

Richard Hawes is a Senior Systems Engineer at the ITSO Austin Center. He writes extensively about the Tivoli Management Environment especially the Security discipline. Before joining the ITSO in July of 1997, Richard worked in the European Software Project Office (based in the UK), where he performed technical troubleshooting and critical situation-management on IBM products for OS/2 Warp Server and Windows NT.

Esau Moises Acevedo Gutierrez is a Services Specialist in Mexico. He has more than seven years experience in IT including managed operations and technical support fields. Moises has consulted for various industries in Mexico including manufacturing, banking, and other financial institutions.

Technical Professionals in services, development, and other areas always have many demands on their time - especially in the rapidly expanding Tivoli arena. We are, therefore especially, grateful to the following people for their invaluable contributions to this project:

Abdul Malik Yoosufani
Kate Brew

Nancy Ball
Rick Fafard
Tivoli Systems

Tivoli Management Product Names

In an effort to eliminate any confusion about the names for Tivoli's expanding line of management products, Tivoli has recently been through a brand naming review. Those already familiar with the products mentioned in this publication will be used to seeing the names as TME 10 User Administration, TME 10 Security Management, and so on.

The new naming convention for these enterprise software management products replaces TME 10 with Tivoli so the new names are Tivoli User Administration and Tivoli Security Management. (This change may seem trivial, but the consistency comes from more dramatic changes on other products, such as Unison Destiny, which is now Tivoli Output Manager).

Throughout this publication, we have endeavored to use the new names wherever practical. This includes references to the Tivoli Management Agent, often referred to in the past as the Lightweight Client Framework (LCF).

Comments Welcome

Congratulation or criticism, your comments are important to us!

Beyond the product manuals, there is very little material available to assist in the design process, and so, our aim was to publish this material as quickly as possible. In this text, we concentrate on Windows NT and UNIX platform support. At the time of publication, Tivoli User Administration also had support for IBM's OS/390 Security Server (RACF) and Novell NetWare, with other endpoints coming soon, such as LDAP and OS/400. Future versions of this publication and "*Getting Started With Tivoli User Administration* - SG24-2015" will add more detail related to all supported platforms. Also look out for a new Redbook due before the end of 1998 - the provisional title is *Managing the OS/390 Security Server with Tivoli*.

We want our Redbooks to be as helpful as possible. If you have implementation experiences that would benefit others or have any comments about this edition, please send them to us in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 131 to the fax number shown on the form.

- Use the electronic evaluation form found on the Redbooks Web sites:

  For Internet users `http://www.redbooks.ibm.com`
  For IBM Intranet users `http://w3.itso.ibm.com`

- Send us a note at the following address:

  `redbook@us.ibm.com`

Chapter 1. Introduction

This chapter gives an overview of a suggested methodology for implementing Tivoli User Administration solutions. This can be viewed in four major phases:

- Gathering information about the environment and detailed customer requirements
- Analysis of the information leading to a design
- The implementation of the design
- The operation and maintenance of the solution

This Redbook was built with the input and experience of many people, and the result is a suggested approach that may apply directly to your situation or that can be a guide to anyone involved in implementing Tivoli User Administration.

In this chapter, we will describe some key points of each phase of our methodology. Throughout this publication, we talk about *users*. This, in general, refers to packets of user account information that defines a real end-user (or principal) to the systems or application we are managing. We configure the user account information through the Tivoli User Administration interface. This configuration information is stored through user and group profiles in the Tivoli management database, and following a distribution to user account management systems, the user information is updated in the system-specific user data files.

Our methodology assumes the typical person implementing it is in some form of services organization and is worded accordingly. Some sections may not apply depending on your role as the one who implements the solution, such as in an environment you already administer.

1.1 Gathering Information

There are two main times during the implementation of a Tivoli solution when information is gathered about the customer environment.

The first is during the sales time-frame when as much information as possible should be gathered about the customer's Information Technology (IT) structure and systems management processes. This enables an assessment of the needs and readiness for a Tivoli solution deployment.

The second period for gathering implementation information is usually after a contract is signed, and it becomes necessary to obtain detailed requirements and information about the environment.

1.2 Analysis and Design

This phase will require the greatest amount of work and creativity. You may think of this phase of the project as the technical design phase. The more focus given to this phase, the easier the implementation will be.

Our aim is to provide methods for completing this phase that will apply to many types of implementation and to offer ideas relevant not only to Tivoli User Administration but also to the Framework and other products.

Each implementation will be different, and Tivoli User Administration implementations often include a lot of customizing. While a lot of what we cover may not apply directly, the discussion and the ideas should still help extend your understanding of the issues involved.

During the analysis and design of an implementation, we will look at logical design issues such as the use of distributed or centralized administration models.

1.3 Implementation

When we discuss implementation, we will provide information about the physical installation and configuration of the Tivoli Framework and Tivoli User Administration.

We will look at typical customizing activities associated with an installation, as well as specific considerations to keep in mind. Tivoli User Administration can be extended to support a wide range of products that need to maintain user data. This topic also covers some examples to provide a general understanding of the types of extensions that are possible.

1.4 Operation and Maintenance

During this part of the redbook, we will discuss the daily operations involved in a Tivoli User Administration environment. Examples include the relationships established with the customer's organization, the on-going process of documentation, product maintenance, and the management process.

1.5 Process Overview

When working with a complex project, it can be useful to break things down into process models. This publication does not address the wide topic of process management and design, but we will present here an example of using process mapping as it relates to understanding user administration. Further examples of process documentation can be found in Chapter 5, “Operations and Maintenance” on page 87.

The remainder of this section provides an overview of how we can begin to manage the concept of a complicated topic (system management) by looking at it in a structured, process-oriented way.

Simply put, the system management process is a grouping of tasks that enables a system management function to be accomplished. A process is a group of activities and actions that people and machines perform to accomplish a system management task.

Since we want to know how the user administration task is accomplished by the customer, we need to interview customer process owners keeping aware of the fact that we may find more than one owner in the same process.

The analysis of a user administration process could show areas where improvements could be made, such as by re-assigning roles, merging or eliminating job roles, or providing input to configure Tivoli User Administration in a different manner than might first have been anticipated.

We are going to describe a way to document the user administration process to further ensure an effective implementation of Tivoli User Administration. It may not be necessary to create a complete process model. However, it is important to understand the flow and maturity of the customer’s systems management processes that directly affect the Tivoli User Administration solution, and that is why we suggest this approach.

1.5.1 Process Relationships

Since a user administration process is not going to be the only one implemented in an organization, it can be helpful to document process relationships so you can review how the user administration process links with other processes. You also can review how you could be affected by changes in another process, or how changes in the user administration process could affect other processes.

An example of what processes may affect our process of server management (that includes user administration) is given in Table 1.

Table 1. Process Relationships - Processes Affecting Server Management

Process Invoking the Server Management Process	Input Received from Invoking Process	Output Returned to Invoking Process
Moves, Additions, and Changes	Service request and complete description	Status of request results
Customer Services	Service request and complete description	Status of request results
Problem Management	Service request and complete description	Status of request results
Change Management	Service request and complete description	Status of request results

Table 2 lists some examples of how activities in our server management process may affect other processes.

Table 2. Process Relationships - Processes Affected by Server Management

Process Invoked by the Server Management Process	Input Sent to Invoked Process	Output Received from Invoked Process
Problem Management	Description of problem	Resolution status
Customer Services	Request of service	Status of request
Problem Management	Request of service	Status of request
Change Management	Request of service	Status of request
Asset Tracking	Request of service	Status of request
Change Management	Description of change	Change approval or disapproval

1.5.2 Roles and Responsibilities

It is important to understand the responsibilities of those involved in delivering or supporting the user administration process. This will help you to define Tivoli User Administration roles and authorizations. Of course, several roles might be performed by the same individual. Some examples of server management roles (that includes administration) and responsibilities are listed here:

- Server Administrator

  The server administrator has overall responsibility for maintaining, installing, removing, and administering (possibly all) servers on the client network. Server administrator responsibilities include:

  - Maintaining all software and data residing on the network servers.
  - Maintaining the hardware required for the network servers.
  - Installing and configuring requested operating systems and software.
  - Removal of servers.

  Plus general responsibilities, such as:

  - Record all actions/status changes in the request record.
  - Provide status as requested.
  - Make recommendations for process, procedure, and tool improvements.

- User/Group Administrator

  The user/group administrator has overall responsibility for adding, changing, moving, and deleting requested user and group access to one or more servers on the network. The user/group administrator responsibilities include:

  - Reset network.
  - Reset user passwords.
  - Add, change, or delete user IDs.
  - Add, change, or delete groups.
  - Data moves (for example, moving files remaining from departed employees).
  - Software access.
  - Space allocations.
  - Name changes.
  - Name access restrictions.
  - Login script adjustments.

- Mail Administrator

  The mail administrator has overall responsibility for maintaining, installing, and removing electronic mail servers and administering user and group accounts for the mail servers. Mail administrator responsibilities include:

  - Maintain and manage the mail servers.
  - Maintain and manage the calendar servers.
  - Maintain and manage all mail and calendar gateways.
  - Provide Level 3 support for the mail services.
  - Provide timely troubleshooting and fixes for the managed mail servers.

- Process Architect

The process architect has overall responsibility for the server management process and procedures. Process architect responsibilities include:

- Own server management procedures for a service delivery organization or account.
- Develop policies pertaining to the organization/accounts.
- Communicate new/changed policies.
- Approve/reject procedure deviation requests.
- Assist server, user, and mail administrators with process/procedure implementation.
- Assist in reassignment of misdirected requests.
- Identify potential noncompliance with service level agreements for handling server management requests.
- Contact server, user, and mail administrators to ensure resolution of potential noncompliance situations.
- Design and produce service level reports package for server management.
- Negotiate and arrange for development/maintenance of account and tool-specific processes and procedures.
- Represent process requirements and enhancements to the process owner.
- Review, approve, and implement operational process updates.

1.5.3 Process Detail

For the following discussion, refer to the diagram of a process flow given in Figure 1.

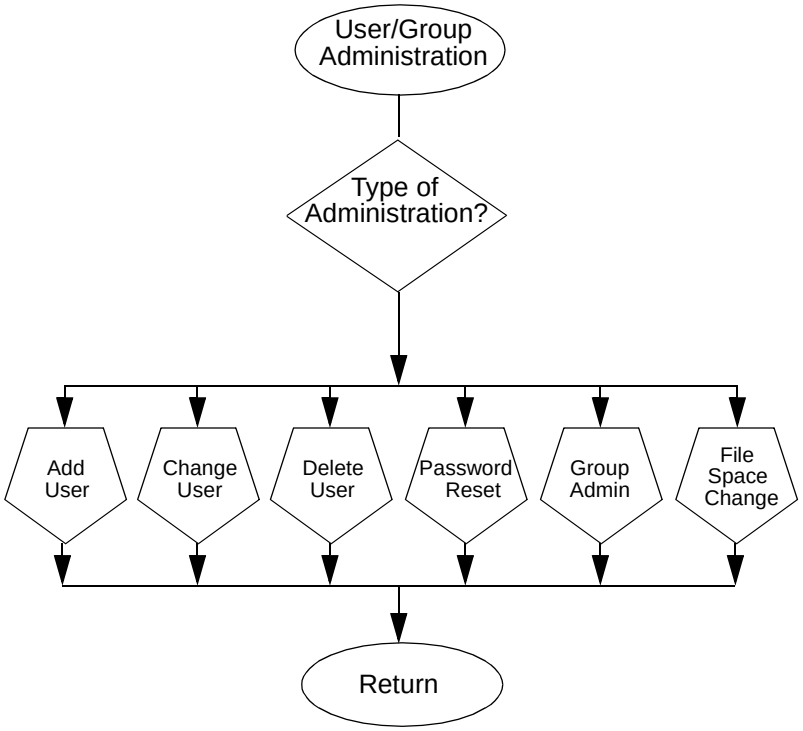


Figure 1. User and Group Administration Process Flow

We will expand on the flow diagram, adding an objective, inputs and outputs from other processes, roles, tools, and prerequisites:

Objective	To administer user and group accounts by adding, changing, and deleting access to the server
Role	User Administrator
Tools	UNIX user management tools
Prerequisites	None
Inputs	Service request from external operational process
Output	Updated documentation and service request record
Subprocess called	Add User / Change User / Delete User / Password Reset / Group Admin / File Space Change
Called by subprocesses	None

The *User/Group Administration Process* includes the following flow:

1. Type of Administration? (Determine the type of administration required by the service request.)
   - If Add User, proceed to Add User.
   - If Change User, proceed to Change User.
   - If Delete User, proceed to Delete User.
   - If Password Reset, proceed to Password Reset.
   - If Group Administration, proceed to Group Administration.
   - If File Space Change, proceed to File Space Change.
2. Add User (Subprocess)
   - Return
3. Change User (Subprocess)
   - Return
4. Delete User (Subprocess)
   - Return
5. Password Reset (Subprocess)
   - Return
6. Group Administration (Subprocess)
   - Return
7. File Space Change (Subprocess)
   - Return

1.6 Tivoli Management Architecture Overview

For the most part, this publication assumes a certain level of knowledge regarding the Tivoli Management Architecture and Tivoli User Administration. The remainder of this chapter serves as an overview of the more important aspects as they relate to a Tivoli User Administration design.

Today's computing environment relies more and more on a distributed client/server model for information system needs. Users at the client workstations effectively use the network as one big server or service provider. Distributed computing or network computing ties people, information, and resources more closely together, but brings a challenge when considering the management of these systems. Managers face the complex problem of maintaining many different types of hardware and operating systems and multiple user maintenance models across those systems.

Tivoli talks about management from the desktop to the data center, managing PCs, workstations, servers, and on up to mainframes. The aim of the Tivoli Management Architecture is to provide a way to manage network computing resources of many different types from a single location through a consistent management interface.

1.6.1 Tivoli Management Architecture and Products

Tivoli includes an open framework that third party management applications can exploit. Tivoli also delivers a set of management applications in key management areas, such as user administration:

- The Tivoli Management Framework provides basic system administration capabilities, as well as a set of foundation services for management applications. These capabilities and services include a Tivoli administrator and administrative privilege facility allowing the delegation of responsibility, a system policy facility, a notification facility, and other services, such as communication mechanisms between management agents and basic scheduling and task-oriented services.
- The Tivoli desktop provides a graphical user interface (GUI) and a command line interface (CLI).

Management applications from Tivoli and other vendors provide tools for managing specific system resources and services. Tivoli includes, among others, applications for software auditing and distribution, remote monitoring, user administration, and internet and intranet management. There are additional tool kits designed to ease integration:

- The Tivoli Application Extension Facility (AEF) enables you to customize Tivoli applications by adding site-specific behavior such as the attributes used or the look of the GUI for otherwise standard, off-the-shelf applications.
- The Tivoli Event Integration Facility (EIF) enables you to build event adapters to map events from any application, resource, or component into a format compatible with the Tivoli Enterprise Console.
- The Tivoli Application Development Environment (ADE) includes programming tools for creating new custom management applications on top of the Framework. The ADE documentation is essential reading if you wanted to become familiar with the internal workings of the Tivoli Management Framework.

The picture shown in Figure 2 is commonly used to show the relationship between applications such as Tivoli User Administration and the other Tivoli components. The management applications all use the framework and make

use of common facilities, such as the management object database and security and communication functions.

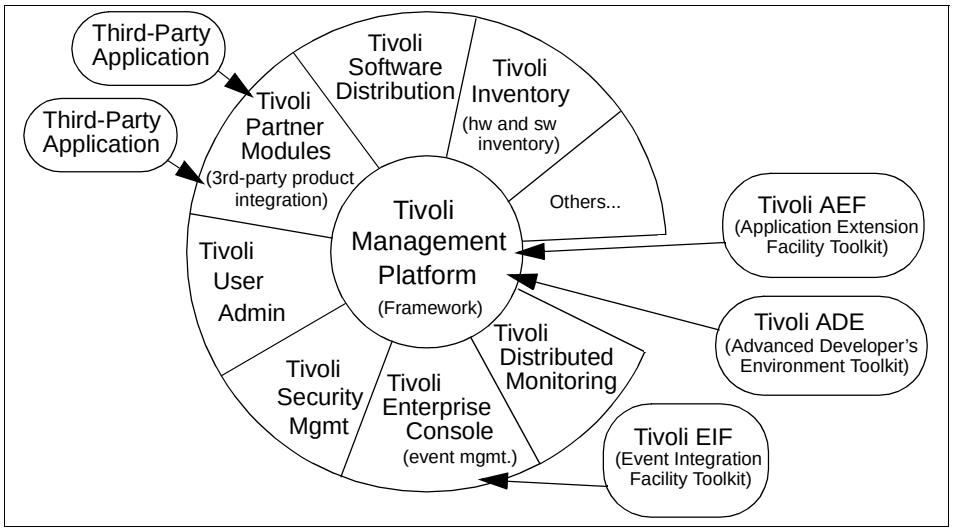


Figure 2. TME User Administration Relationship with Tivoli Components

Partner applications integrate at varying levels. Usually, integration will include items, such as some use of the Tivoli desktop, installation and distribution of programs through the framework and some form of event integration.

1.6.2 The TMR Server and Clients

The Tivoli Management Architecture, in general, refers to the set of Tivoli management applications, and Tivoli Management Environment is often used to refer to a set of Tivoli products functioning at a particular site.

The Tivoli Management Environment consists of Tivoli Management Regions (TMRs) each of which will have one TMR server. A TMR server runs software and a database of management information that allows it to manage Tivoli clients. The Tivoli clients run software that allows them to interact with the server. A Tivoli client is defined and managed by one server.

1.6.2.1 The Tivoli Management Region

The basic unit of Tivoli functions is the Tivoli Management Region (TMR). It consists of one TMR server and the clients that the server is managing. The server for a Tivoli Management Region is normally referred to as the TMR

server for the TMR and will hold the management information database for that TMR.

Depending on the size and requirements of an environment, there may be more than one TMR defined. If multiple TMRs are present, they can either stand alone or be linked together. Linking TMRs allows management functions to be exchanged between the regions. These connections can be one-way or two-way in terms of access permissions and exchanging information between TMRs.

After you connect two TMRs, you should schedule periodic exchanges of resource information between the TMRs. This exchange includes information on application-services resources (policy regions) and application-defined resources (users, for example).

Once you have connected TMRs and exchanged TMR resource information, you can manage resources in a remote TMR from a connected TMR. You can also disconnect TMRs at a later time if desired.

1.6.2.2 Tivoli Clients

The Tivoli environment can include three types of clients (also often referred to as endpoints):

- Managed nodes
- Tivoli Management Agent
- PC managed nodes

A managed node runs the full Tivoli Management Framework software and can perform the same security and communication functions performed by the TMR server. A managed node maintains a client-specific database, which is significantly smaller than the TMR server database. A managed node can also be a proxy system for a PC managed node, an endpoint gateway to support Tivoli Management Agents, and a NetWare managed site.

The Tivoli Management Agent is likely to become the most common type of machine in most Tivoli installations. This machine is not used to perform day-to-day management tasks. Instead, it is just one of the machines to be managed. The agent is a small amount of Tivoli software that does not maintain a local database. Each Tivoli management product provides the methods necessary to implement their function, and these methods are downloaded to the endpoint at the time they are required.

All Tivoli management products introduced Management Agent support in release 3.6. The Tivoli Management Framework first implemented the support

for Management Agents in version 3.2 (at which time it was referred to as Lightweight Client Framework - LCF). In future releases, the Tivoli Management Agent and the Managed Node will be the only supported endpoints.

A PC managed node (PCMN) was Tivoli's original solution for managing clients that did not require the framework. The PCMN is an object on a managed node acting as an interface to a PC running some PC agent software. You must install the PC agent software on the endpoint and create a PC managed node in a managed node, if you need to manage PCs running operating systems other than Windows NT, and you are using older applications that are not yet enabled for use with the Tivoli Management Agents. The PCMN implementation limited the functionality of endpoints to what could be provided in the PC agent software.

1.6.3 Tivoli User Administration Overview

System administrators can spend a considerable amount of time managing user and group accounts in a distributed enterprise. User accounts are constantly being added, removed, and updated as users' needs change. Traditionally, the system administrator had to deal with user accounts on only one operating system, such as on a mainframe or on a server, such as UNIX, Windows NT, or NetWare. Today, most distributed enterprises have users that need accounts on any and all of the above architectures. Each operating system's user account management architecture has its own set of configuration files or databases that describe the user accounts for that system. Also, each variant of UNIX has its own implementation of configuration files that describe the user accounts. As well as dealing with the differences in user accounts on each operating system architecture, there are inevitable variations in user account management across systems and organizations within an enterprise.

Tivoli User Administration provides you with the tools necessary to manage user accounts on all the major operating system types.

1.6.3.1 Users and Groups

Tivoli User Administration is a framework profile-based application that runs on your TMR Server with a component installed on the systems that manage the user accounts. It manages user accounts across the network, as well as group accounts and group memberships.

Using Tivoli User Administration, you are able to populate a user or group profile with records from existing account servers. Updated profiles can then be distributed to supported endpoints. Possible endpoints include:

- Other profile managers (to implement hierarchical management)
- Endpoints installed on account servers:
  - NIS domains
  - Windows NT domains (PC managed nodes and NT managed nodes)
  - Windows NT Workstations (PC managed nodes and NT managed nodes)
  - NetWare NDS trees (PC managed node)
  - NetWare 3.X servers (PC managed node)
  - OS/390 Security Server (Tivoli GEM required prior to Tivoli User Administration version 3.6)
  - Other endpoints, such as AS/400, IBM Global Sign-On and Lotus Domino, supported through add-on products (see 4.3, “Broadening Supported Platforms” on page 76)

1.6.4 Profile and Profile Manager Review

A profile is a standard framework object, a collection of information related to a specific application that lets you manage a particular type of resource. A profile also contains a list of subscribers (members of the list) to which the profile can be distributed in order to update system configuration information.

There is a strong relationship between profiles, profile managers, policy regions, and endpoints. Here are some key points about this relationship:

- Profile managers are created within a policy region. The policy region being a way of delegating management authority in a TMR.
- Profile managers contain profiles and a list of subscribers.
- Subscribers can include managed nodes, Tivoli Management Agents, PC managed nodes, NIS domains, S/390 connections, and other profile managers.

  Having a profile manager as a subscriber allows the administrator to distribute a user profile to that profile manager. Then, another administrator can edit and customize that user profile before distributing it to the target machines. This is a way to delegate responsibility to other administrators within a region.
- Profile manager hierarchies are created when profile managers have other profile managers in their subscriber lists.
- Profile manager hierarchies allow you to manage your resources with a hierarchical delegation of management in your organization.

For more information about profile policy, profiles, and profile managers in general, see the *Tivoli Management Framework User’s Guide*.

1.6.5 Tivoli User Administration Profiles

There are two types of profiles in Tivoli User Administration, user profiles and group profiles. Tivoli User Administration uses user and group profiles to manage user and group account details. When the user or group record information defined in the profile is distributed to its ultimate destination (a profile endpoint subscribed to the profile), Tivoli User Administration can modify the system files, maps, or databases of the endpoint to reflect the information defined in the records. Endpoints can subscribe to and receive distribution from user and group profiles across the boundaries of Tivoli Management Regions (TMRs) - assuming the TMRs are connected in the right direction and have exchanged the necessary resources (either through the GUI or using `wupdate`).

A user profile is a collection of user records. Each user record contains general user information, as well as information concerning one or more platform-specific account types (such as UNIX, Windows NT, OS/390 Security Server (RACF) and NetWare) for a single user. A group profile is a collection of group records. Each group record contains all the information needed to define UNIX group accounts. This includes information such as the group name, the group identification number (GID), and a set of the members' login names. The Windows NT and NetWare user account records are used to manage group membership (Windows NT Group management is a function of Tivoli Security Management mainly because it is more closely aligned with the Tivoli Security Management role-based management approach).

1.6.5.1 Profile Policies

Policies are rules that users and groups must comply with. For example, the administrator might want all user passwords to have a minimum of five characters (an example of password policy). These policies allow the administrator to keep all the user and group definitions (attributes) consistent across all platforms within a region. There are two types of policies used in a profile: default policies and validation policies.

Default Policy

Each profile has a default policy associated with it. The default policy determines the default values used when creating a new entry in a profile (for example, when creating a new user in a user profile). These default values help you minimize the amount of data that you have to enter when creating a new record in a profile. They work as a template for each new record you add to the profile. Default policy provides a value if you don't fill in the blanks. A default policy ensures that all users have a set of consistent default values for their attributes.

Note

For performance and other reasons, populating a profile does not run the default policy. If you populate users from a UNIX system, Windows NT details are not generated for the user record (as they would be if you entered UNIX details in the GUI and then hit **Generate Defaults**).

There may be times when you would want the default policy run against a populated record. Tivoli plans enhancements to the `wpopusrs` command or a new command to enable the default policy to be run optionally at populate time. Currently, the alternative is to open the user record in the GUI and hit the **Generate Defaults** button. Other methods have been used for larger numbers of user records. For example, you could write a script to parse the /etc/passwd file and feed it into `wcrtusr` as an alternative to using `wpopusrs`.

Validation Policy

Every time you modify or create a profile entry, it is checked against a set of validation policies that ensure that the data you are entering complies with the current policy. Validation is performed in the same way by default when populating a profile. This prevents you from creating or getting an entry that does not meet your specifications. You can also request a validation for a specific profile at any time either through the profile GUI or by using the `wvalidate` command.

The policies can be edited from the GUI or from the command line interface. Each policy attribute can be set to a constant, to a script, to a regular expression, or to none. Refer to the Tivoli User Administration manuals for more information and some examples.

Default and validation policies are stored in the profile and can be set for any attribute on any profile or profile copy. This allows different policies to exist on the top-level profile than on the copies of the profile held at the managed node.

1.6.5.2 Profile Population

Populating a user or group profile consists of gathering all user and group information from the system files on the endpoints and adding or merging that information to the profile records.

Tivoli User Administration user and group profiles can be populated from sources, such as a managed node, PC managed node (Windows NT or NetWare), or OS/390 connection.

User and group profiles are managed resources, and each policy region maintains a list of managed resource types that are valid for that specific policy region. In order for an administrator to create or manage a profile, the following must be true:

- The profile manager must be a managed resource of the policy region.
- The particular profile type, such as user or group, must be a managed resource of the policy region.
- Administrators must have the senior role if they are to create profile managers.
- Administrators must have the appropriate roles depending on the type of administration they will be performing on the profiles in the policy region.

When populating a profile, the validation policy for that profile applies. This means, for example, that UIDs less or equal to 0 will fail the validation. In UNIX, users root and nobody will not pass the validation policy.

1.6.5.3 Profile Distribution

Once a profile is completed, by either a populate or by adding records, it can be distributed to the subscribers.

When distributing a profile, there are four options: Next Level, All Levels, Preserve local modifications, and Exact copy. It is extremely important to understand the differences between these options. These options are covered in detail in at least two other redbooks: *TME 10 Internals and Problem Determination*, SG24-2034, and *Getting Started With TME 10 User Administration*, SG24-2015.

1.6.5.4 Profile Synchronization

If the system files and databases of a profile endpoint are changed directly without using Tivoli, the profiles that the endpoint subscribes to may no longer accurately reflect the endpoint's current configuration. That is, the users and groups defined in Tivoli User Administration profiles may not reflect the true definitions of the users and groups on the managed systems.

A synchronization function is available to reconcile any differences between the Tivoli profile databases and the current profile endpoint configuration (except for passwords).

When synchronizing profiles, Tivoli will first present a dialog, titled Profile/System Discrepancies, outlining the differences found. The following differences may be found:

Delete Record	Profile items that exist in the profile, but not in the endpoint system files
Add Record	Items that exist in the endpoint system files, but not in the profile
Change Record	Items that differ in the endpoint system files and in the profile

Chapter 2. Gathering Information

This chapter describes the first step in developing a user administration solution. It has been divided into three sections. First, we will determine what the company expects from user administration design, what the aims are. We then talk about the sources of information that could be used to gather the needed information. In 5.5, “Management process” on page 97, we discuss the topic of process-based systems management and look at the interrelationship of different processes with the management of users.

You will find here guidance on how to take a snapshot view of the environment to be administered. It should lead to the following outputs:

- A document showing the objectives of the company concerning user administration
- A company user administration policy document
- A document describing the computer environment and any existing user administration

It is generally accepted that if the designer does not understand the IT structure, or the systems management processes are not well defined, then the solution deployment takes longer than proposed.

2.1 Information Gathering Aims

The first step in creating a good design for any user administration environment is to be sure about the requirements of the customer. Sometimes it can be difficult to obtain this information, even the customer may not really know what their requirements are, and in these cases, understanding the aims can be essential.

Different environments will need different approaches and designs. Solutions may vary from customer to customer to fit specific needs. So you must be sure to obtain the right information and clearly understand the customer requirements.

The aims of gathering information could be listed as follows:

- Understand very clearly the needs or system management requirements.
- Obtain and understand the current IT structure. This includes:
  - Organizational structure, such as areas, departments, and units.

    - Roles and responsibilities within the organization, such as the Chief Information Officer (CIO), managers, coordinators, and administrators.
    - A feel for the customer's culture and policies gained by interviewing as many people as is reasonable, and if possible, spending time at the location.
    - Identification in organization structure of groups involved in systems management.
    - Identification in organization structure of groups involved in the deployment of the Tivoli solution, that is, the people with whom you will be working.
    - Physical location and organizational relationship of the above groups.
    - Business needs that affect the IT structure.
- Determine the maturity or level of definition of each systems management process relevant to the user administration solution.
- Understand the managed environment, including all managed resources and current management tools in place.
- Determine the customer's perception of the efficiency of their current processes.
- Understand whether the customer is able and willing to change some of their processes with the new user administration solution.
- Identify additional opportunities to further automate management based on their current processes.
- Determine the factors that might affect the deployment of the user administration solution, such as future policy changes due to re-engineering, mergers, or acquisitions.

2.2 Information Sources

We need to determine who will be able to provide the information we need. The IT department is most often a logical starting point. Part of planning an implementation project should include coordinating the availability of resources necessary to complete each step at this time. The ordering and importance of the interviews is generally arbitrary. Most environments will have time constraints that cannot be predicted in a document of this type, causing you to reorder the steps.

There may be a large number of interviews to do in order to obtain all the information required. The suggested approach involves reviewing information gathered by others, such as sales people, in advance, then creating a schedule of interviews with the people best suited to providing answers.

Depending on the size of the organization, you might find these people entirely within IT, or you may find them in separate departments outside IT, with IT serving as the go-between.

Before you go through interviews, you should review any other available information you can, for example:

- Talk with the sales team about the customer service perception, business needs, and the implementation time-frame expected.
- Review previous requests for proposals made by the customer, if they exist.
- Identify any other services delivered to the customer and talk with the people involved in those services to obtain their perception about the customer environment.

Now that you are prepared with all that basic information about the customer requirements and needs, the following is a suggested a list of people with whom you should discuss the implementation, documents to review, and information you should obtain.

2.2.1 Project Manager

The common approach to dealing with a services project is for the organization to appoint a Project Manager who will be involved in all the implementation processes and will later be responsible for the services deployment at the customer site. This may also be a role you are taking on.

The information expected from this person includes:

- Description of customer business
- Customer business needs
- Organizational structure
- Customer expectations about the Tivoli User Administration solution
- User administration process owners that will be involved in the solution
- User administration policies or standards
- Customer environment and culture

2.2.2 User Administration Process Owners

User administration process owners may be dedicated administrative staff, or the process ownership could be split among different people who have other roles in the organization but own parts of the administrative process, such as

giving access to resources to the users and doing other user administration activities.

The information expected from these people includes:

- Explanations about the overall administration process
- Documented information about any user administration processes in place
- Tools used in these processes
- Expectations for the Tivoli User Administration solution
- Process improvement expectations
- Resources involved in the current process
- Names of related process owners

2.2.3 Previous Analysis

Sometimes, the customer already has information about previous system management process consulting available for review. If the authors are available for a discussion of their findings, this would be the best approach, but in any case, review any work already performed. Information that can be gained from prior projects includes:

- Organizational structure
- User administration process owners
- User administration policies or standards
- Process descriptions
- Documented information about any user administration process in place
- Resources involved in the current process
- Names of related process owners
- Recommendations based on current technology and processes to improve customer proficiency

2.2.4 IT Department

Your understanding of existing IT information should be as accurate as possible to design an appropriate Tivoli User Administration solution.

The information in the following sections should be drawn from all possible sources.

2.2.4.1 Naming Conventions

Users and applications usually access computer resources or objects, such as computers, shared directories, and printers, by using the name assigned to the resource. If there is a naming convention in place, where the names are consistent for all variations of resource types, it helps the user to select quickly the proper resource for their work and simplifies administration.

Several questions must be answered in the context of naming conventions:

- Which organization is responsible for defining the naming convention?
- What names are given to existing resources?
- Does the name have a structure? Identify each part and the delimiting character used for separating parts.
- Is there a mechanism to validate the uniformity of names for newly defined resources? In Tivoli, we could use validation policies to enforce this.
- Are there any dependencies with other naming conventions?
- Are there limitations, such as characters used or the length of names?
- Are there platform-specific naming conventions?

The following commonly used objects in a computer environment should be considered when asking about the naming convention. These objects are:

- TCP/IP names
- Microsoft-style computer and domain names
- User and group names
- Tivoli enterprise software objects

TCP/IP Names

A TCP/IP name identifies a computer that can be connected to/from another machine with the TCP/IP protocol. It is easier for the user to address a computer with the TCP/IP name instead of the IP address.

- It should be possible to obtain a list of machines and their names.

Microsoft-style Computer and Domain Names

The Microsoft-style computer and domain names are applied if you use windows applications to access the network through the NetBEUI protocol (the NetBIOS extended user interface). NetBIOS is the standard API for Microsoft network products.

These names are necessary to know if the resources of a computer will be shared with many end users. Typically, a user will browse the network looking

for the name of a server known to contain a particular resource, or the user will directly connect to a share from a given server if the name and resource name are known.

User and Group Names

Every user logged into a system has a login name that is typically assigned to one or more system-specific group names. The users' membership to a list of groups typically defines access rights to resources, because the access to resources is commonly granted to groups, rather than individuals.

- Check the organization structure.

  Quite often, companies use a structure for user and group names to consider the organizational relationship.

- Equal user names on all platforms.

  Should/can a user have the same login name for all supported platforms?

Tivoli Enterprise Software Objects

If Tivoli products have already been installed, there will hopefully be a naming convention for all current resource types. Each product may have different resource types, for example:

- Framework

  TMR, Generic Collection, Administrator, Policy Region, Subregion, Managed Node, Endpoint, Profile Manager, Task Library, Task, Job

- Distributed Monitoring

  Sentry Profile, Indicator, Indicator Collection

- Tivoli/Enterprise Console

  Event Source, Event Group, Event Classes

- User Administration

  User Profile, Group Profile

- Security Management

  Security Profile

2.2.4.2 Tivoli Environment

If the company already operates with Tivoli, we need full documentation of the existing management environment. You can start the investigation with the following questions:

- What Tivoli product components are in use?
- What version has been installed?

- Who are the responsible product managers?

  These managers or delegated staff should be able to answer your questions about the Tivoli environment.

For this assessment, we will consider only Tivoli products that are relevant for user administration issues. These products are:

- Tivoli Management Framework
- Tivoli User Administration
- Tivoli Security Management

Please remember that the commands listed in the next chapters can only be executed if the Tivoli environment has been set. The environment can be set with these shell scripts (paths shown are the defaults):

- `/etc/Tivoli/setup_env.sh` for UNIX platforms - default shell
- `\WINNT\system32\drivers\etc\Tivoli\setup_env.cmd` for NT platforms

Tivoli Management Framework

As you should know, the Tivoli Management Framework, or in some cases, the Tivoli Management Agent is a prerequisite for the Tivoli products below. It must be installed on user account machines in order to manage user accounts with Tivoli User Administration.

Find out how many Tivoli Management Regions (TMRs) are installed. Each server can have several one- or two-way connections to other TMRs. These inter-region connections are responsible for the exchange of management information between them. It is a good idea to show these dependencies of the TMRs in a map.

Table 3 provides a suggested list of information to gather for each TMR server:

Table 3. Collecting Information: Tivoli Management Framework

Information	How To Get the Answer
Directories used and the file permissions of the Tivoli database, binary, and library files.	On UNIX, use this command: `ls -ld $BINDIR $LIBDIR $DBDIR`
	Use the Windows NT Explorer on Windows NT platforms to determine the directory access rights for results of: `echo %BINDIR% %DBDIR%`
Determine the password to install products on the TMR server (if in use).	It was typed in during the installation.

Information	How To Get the Answer
Check if remote client login allowed	`odadmin odinfo 1`
Check if Kerberos is in use	`odadmin odinfo 1`
Encryption level for communication between object dispatchers within the local region	`odadmin odinfo 1`
Encryption level for communication between Tivoli servers	Tivoli Management Framework will use the level of the target server to communicate with it.
List of inter-region connections to Tivoli servers	`wlsconn`
List of Tivoli Administrators and their properties	`wlookup -r Administrator -a` `wgetadmin <administrator>`
List of managed nodes	`wlookup -r ManagedNode -a`
List of profile managers to which each managed node subscribes	`wlssub @ManagedNode:<label>`
List of task libraries and their contents (tasks and jobs)	`wlstlib -a`
Note: It is recommended that the commands be used on each TMR. The `wlookup` will list objects known to all connected TMRs but only if the resource exchanges are up to date.	

Tivoli User Administration

If Tivoli User Administration is already in use anywhere, Table 4 shows the suggested information to gather.

Table 4. Collecting Information: Tivoli User Administration

Information	How to Get the Answer
Custom categories and attributes for user records	Ask the administrator and/or use: `wlsusrcat` and `wlsusrsubcat`
Default policies for user and group profiles	Select Edit -> Default Policies in the GUI or use `wlspol -d`
Validation policies for user and group profiles	Select Edit -> Validation Policies in the GUI or use `wlspol -v`
Distribution actions	Ask the administrator or use Profile -> Distribute Defaults in the GUI

Information	How to Get the Answer
Assignment of users/groups to profile manager(s) and their policy regions	Ask the administrator
Name of user and group profiles	`wlookup -r UserProfile -a` `wlookup -r GroupProfile -a`
Subscribers of user and group profiles	`wgetsub <profile manager>`
List of user records of each user profile	`wlsusrs`
List of group records of each user profile	`wlsgrps`

Many attributes of a user record have security-related aspects. Therefore, identify the security policy for the following user attributes:

- Time restrictions for login
- Password aging
- Audit control

Tivoli Security Management

This product is responsible for assigning access rights to computer resources for user accounts through security group associations. It is a logical extension to the management of user account information with Tivoli User Administration and can also be employed in isolation. If it is deployed, we should gather the information listed in Table 5.

Table 5. Collecting Information: Tivoli Security Management

Information	How To Get the Answer
List of security profiles	`wlookup -r SecurityProfile -a`
Attributes of each security profile	`wlssec <security profile>`
Default policies for user and group profiles	Select Edit -> Default Policies or use `wlspol -d`
Validation policies for user and group profiles	Select Edit -> Validation Policies or use `wlspol -v`

2.2.4.3 System Policies

System policies define the rules for identification and authentication of the users to the systems of the company's environment and set the standard or default access to the business resources.

In order to design a system policy, we need to know the current policies that are implemented in the environment. This section contains tables that can be

used as forms to fill in the required policy for different subjects. Note that these tables are for Windows NT and UNIX. You may wish to adapt these tables for other platforms, or at least, use them as a guideline for the areas that need consideration.

If there is some default, UNIX user policy can be collected as detailed in Table 6. If the policy varies for different users, you will need to understand the rules in place.

Table 6. UNIX User Policy

Attribute	Selection
Administrator or user controls password	
Is password pre-expired on creation?	
Password aging (min/max life span)	
Home directory local to host or remote	

If there is some default, Windows NT policy can be collected as detailed in Table 7.

Table 7. Windows NT User Policy

Attribute	Selection
Account type (User, Admin, or Guest)	
Login script	
Login times	
Password required?	
Can user change password?	
Force user to change password on next login	
Account expiration in use?	
Allowable login workstations	

2.2.4.4 User Groups

The resources are assigned to user groups, which in turn define proper access rights. We, therefore, need to know which groups exist. For every group, the following questions must be answered:

- What function do members of the group represent?
- Who are the members of that group?
- On which computer is a group defined?

Identifying which group users belong to can be tricky. We suggest the following examples of approaches for AIX (which may or may not apply to other UNIX variants) and Windows NT:

- On Windows NT, the tool `findgrp.exe` is a part of the *Windows NT Server Resource Kit.*
- For AIX, systems there is the command `lsgroup ALL`.

If the company already uses Tivoli User Administration to manage user accounts, it is much easier to get the group memberships. The command `wlsgrps` delivers the group names of the entire company and the users that belong to them. The following is a way of using this:

```
wlookup -r GroupProfile -a |
while read GROUP_LABEL GROUP_ID
do # Get a list of group names and users that belong to

   wlsgrps -l @GroupProfile:$GROUP_LABEL
done
```

Table 8 is an example of the appropriate information to gather for groups.

Table 8. Group Attributes

Attributes of a User Group	**Values**
Group name[1]	`system`
Description[2]	`Maintaining critical system resources`
Name of the system where it resides[3]	`rh2430b.itsc.austin.ibm.com`
Operating system[4]	`AIX 4.2.0.0`
Responsible job role[5]	`System Administrator`
Group members[6]	
User members[7]	`root`
1 Name the group. 2 Describe the job function of the group members. 3 On what platform is the group defined. 4 Enter the operating system and version. 5 Identify the job role responsible for maintaining the group, such as assigning users. 6 In the case of Windows NT, a local group may have global groups as members. 7 Users that are members of the group.	

Chapter 3. Analysis and Design

The process of analyzing and designing a Tivoli User Administration solution under the Tivoli Management Architecture (TMA) is going to be fundamentally based around the information technology resources already in place. In addition, the architecture needs to be flexible enough to provide for expansion of the Tivoli User Administration model by adding administrators, profiles, and users as the need to support more computing resources arises.

The purpose of this chapter is to provide some suggested methods and ideas related to the design of a Tivoli User Administration solution.

The chapter is broken down into a number of areas. The design introduction gives an overview of the topic and explains the significance of creating good designs. Then, we look at a review of the Tivoli Management Architecture and Tivoli User Administration - skip these sections if you are already very familiar with these topics. In the area of logical standards, we define the use and application of the Tivoli User Administration configuration options, and for physical standards, we look at topics such as the machine requirements necessary for proper installation and operation. The last piece is devoted to additional considerations for the design.

3.1 Design Introduction

The analysis and design phase of a user administration solution is an important step. The more focus given to this phase of the project, the easier the implementation will be. This phase is the transition and thinking process that happens between the collection of detailed requirements/information and the final solution design.

It makes sense to involve the customer periodically during the design development to ensure all management aspects are being covered.

Implementing a new user administration solution is often likely to require individual customizing, special considerations, and so on. There is unlikely to be one perfect way to explain how to get to a comprehensive Tivoli User Administration design, but the following sections describe a suggested approach that is based on real-life implementation experiences.

3.2 Logical Design

Even with all the information obtained from the customer, generating a design is a difficult process. A good design relies on experience of Tivoli User Administration installations. In this section, we use the experiences of many who have implemented Tivoli User Administration in the past, reviewing different scenarios and environments and suggesting two approaches to user administration design - centralized and distributed. Which of the two approaches to use depends on the systems management strategy and the physical environment to be managed.

Centralized Single administrator or central administration team

Distributed Site-based administration or a hierarchy of delegated distributed administration

There are different variables to take into account when building a design. These include likely TMR server load, the management of security-related information, and the way resources are shared across the network. The *Tivoli Framework Planning and Installation Guide* gives information regarding the suggested use and requirements of TMR servers. You will need to review both the centralized and distributed administration sections in order to understand the issues involved and build from these descriptions a model to suit the customer.

3.2.1 Centralized Administration

Centralized Administration means having administrators in a single location administering your system resources, even if those resources are spread across multiple physical sites - each site maintains little or no administrative capability. We will review how the physical layout, policy region design, and profile manager works with this concept. The aim is to give one administrator or central group of administrators responsibility for the entire user management environment.

3.2.1.1 Physical Layout

When talking about determining the need for centralized user administration, it is important to note that whether you have centralized or distributed administration, the physical layout of the management system could still be centralized or distributed. This means that even if you have centralized roles in your user administration design, you still may have single or multiple TMRs.

Now, we will review how this centralized view of administrator roles is used in centralized or distributed physical installations. Figure 3 shows two different physical installation layouts.

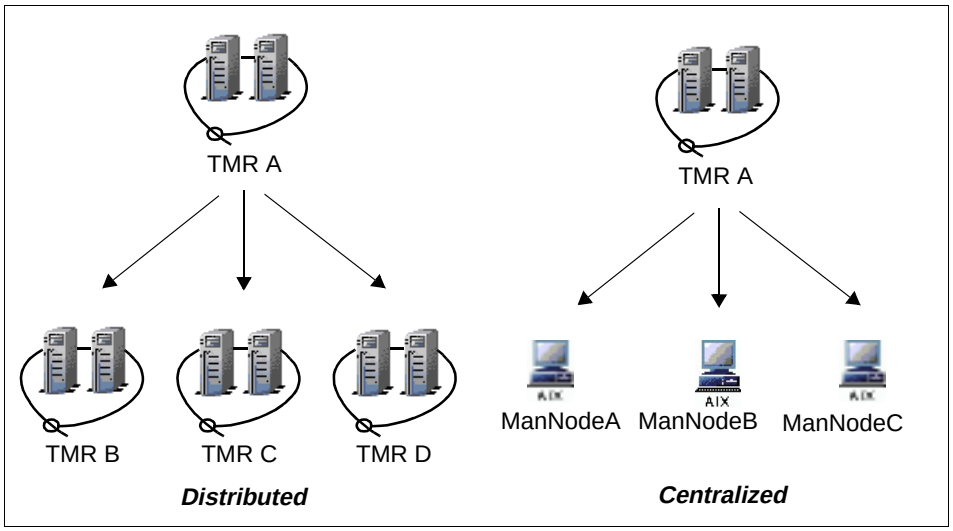


Figure 3. Physical installation layouts

Distributed Shows a configuration, where one system acts as a hub (TMR A in Distributed), and the others are all spokes fanning out from the hub. This configuration can distribute server load across TMRs and handle more clients and may be necessary in environments where the systems to be managed are spread across multiple locations. At the same time as TMR A has a connection to each of the other TMRs, administration could be handled centrally with the majority of the administration work initiating from TMR A.

Centralized Shows a configuration where one TMR server communicates with all the endpoints to be managed.

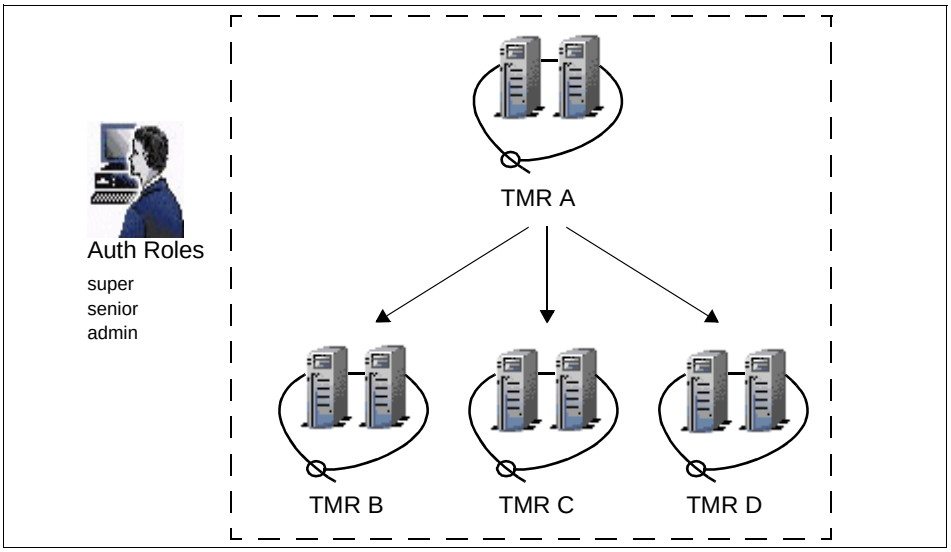


Figure 4. Centralized Administrator Roles - Distributed Physical Layout

Figure 4 shows how an administrator's Tivoli authorization roles (super, senior, and admin) apply in the physical layout. The dotted line represents the scope of control such an administrator has in a centralized configuration. As you can see, the same administrator(s) have their roles applying across TMRs in different locations.

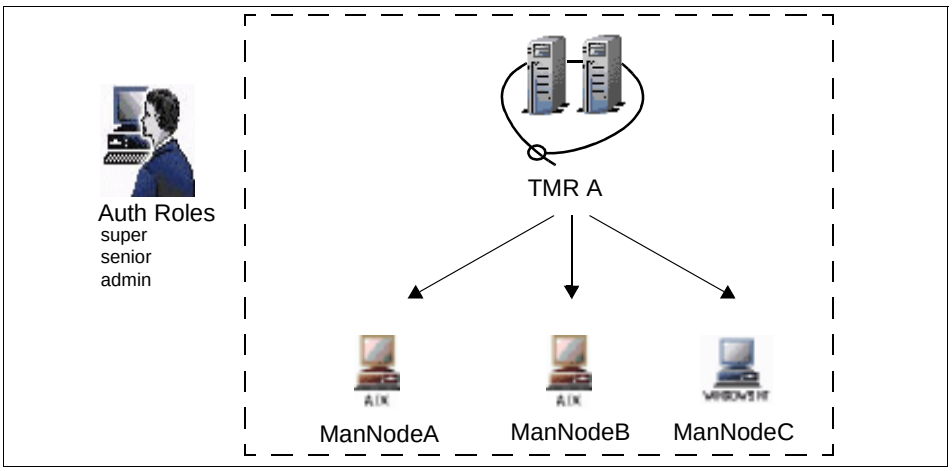


Figure 5. Centralized Administrator Roles - Centralized Physical Layout

Figure 5 shows how centralized authorization roles apply equally to a centralized configuration. Administering everything from a single TMR can impose restrictions on the number of endpoints that can be handled, especially when relying on managed nodes and in implementations where PC Managed Nodes are still in use. Single-TMR implementations can be much larger with the use of Tivoli Management Agents. If you wish to use centralized administration, this centralized physical configuration is likely to be the simplest to implement.

3.2.1.2 Policy Regions and Profile Managers

In order to manage user administration, we have to define a set of policy regions and profile managers in the Tivoli environment. We will define a hierarchical structure consisting of policy regions, sub regions, and profile managers that are able to arrange the profiles in a logical way, and we will also see how they work with centralized and distributed user administration approaches.

We will start by describing an example scenario in order to have a better understanding of the use of policy regions and profile managers. This scenario looks at centralized management of widely distributed systems.

Suppose we have a customer with four region offices. They are Austin, New York, Los Angeles, and Tampa Bay. These provide natural geographic boundaries that would divide the enterprise into usable sizes for TMRs.

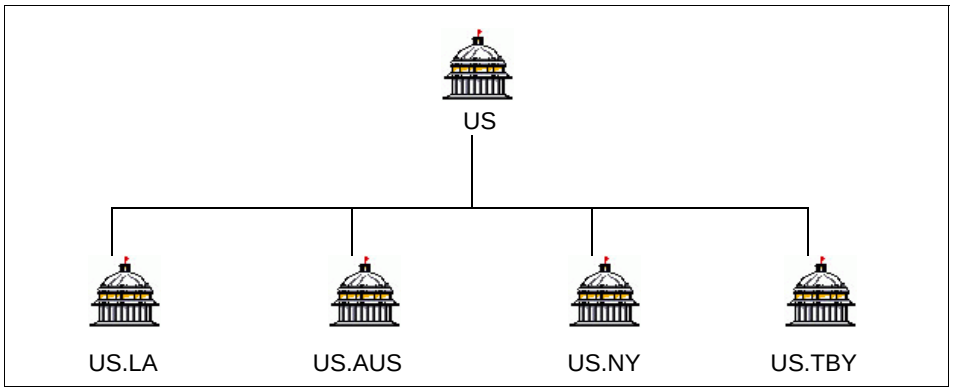


Figure 6. Centralized Administration - Top Level Policy Region Design

Figure 6 shows how we can start to build a hierarchy of policy regions. The U.S. policy region will contain a sub-region used to manage the resources of each of the TMRs. This definition of regions will either take place in a new hub TMR or on one of the regional TMRs that will become the hub. Next, we

will add policy regions that will help us to divide the environment by management-function areas. The examples we add in Figure 7 are a user administration policy region and a software distribution policy region.

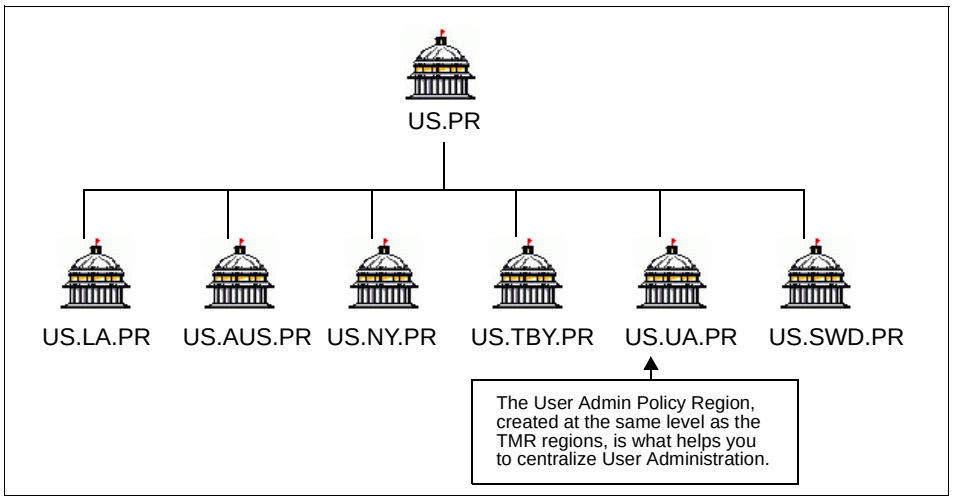


Figure 7. Centralized Administration - Functional Area Policy Regions

Now, we will take a detailed look at one region, the Austin policy region, and see how it is further defined (for this example, we will just expand two policy regions under Austin, one to determine policy for the nodes and the other for user management; however, it could have many more).

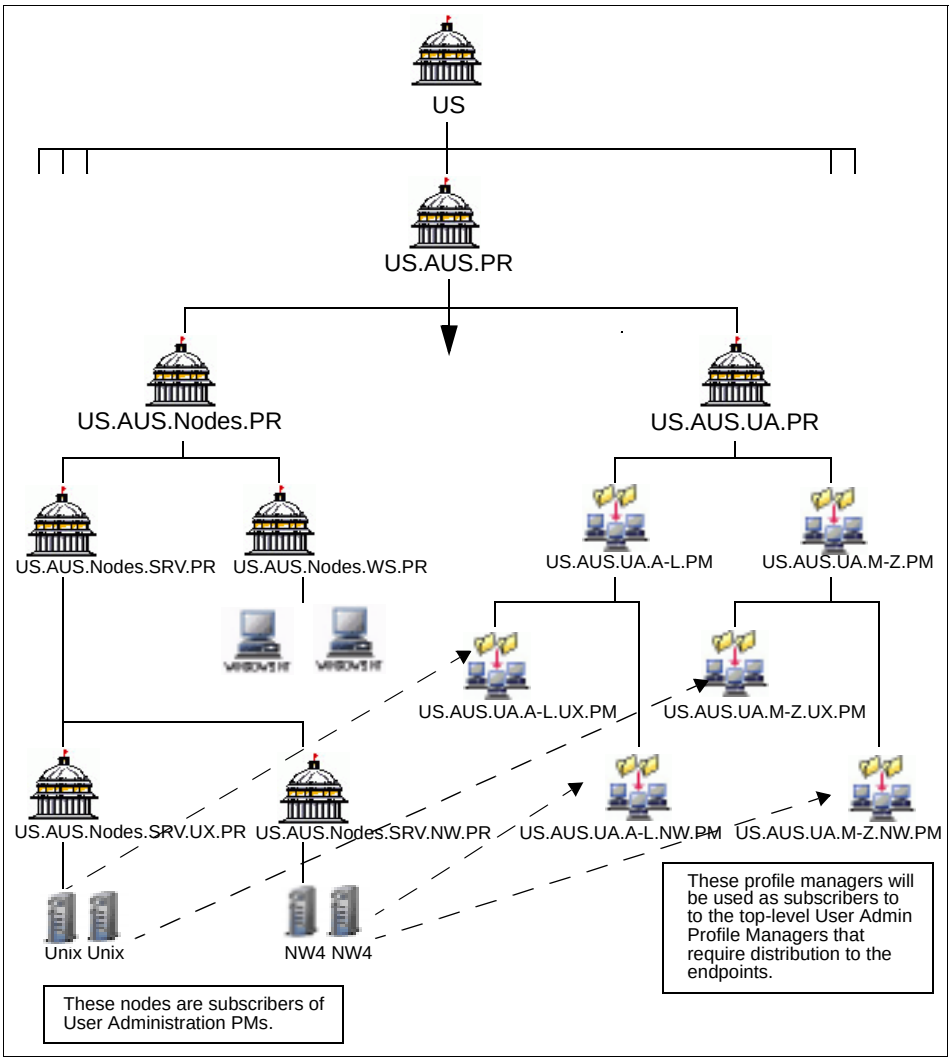


Figure 8. Site-Specific Policy Region Design

As you can see from Figure 8, the Austin policy region contains a nodes policy region and a user administration policy region. In order to allow us to define each US region as an area of responsibility, a subpolicy under the regional top level has been created. All managed system resources of the Austin region will be contained within this policy subregion. This includes the definition of all workstations and servers that are part of the Austin region.

Each functional-area policy region at the first level (in our example, user administration and software distribution) will have a matching subregion in the Austin, Tampa Bay, New York, and Los Angeles regions. The profile managers in those regions (such as US.AUS.UA.A-L.UX.PM) have subscribers of a particular machine class (such as workstations or servers). They won't have any profiles defined and won't have any additional layer of profile managers as subscribers. The subscription list will need to be created and maintained by administrators responsible for the location. Each addition and deletion from the nodes policy region should be reflected as a subscriber addition and deletion in the appropriate profile manager. For example, if a new UNIX server is added to the Austin region, the server will be defined within the US.AUS.Nodes.SRV.UX.PR policy region under the nodes policy region and will be added as a subscriber to the US.AUS.UA.A-L.UX.PM and US.AUS.UA.M-Z.UX.PM.

The picture we have drawn here is about as complicated as a design should be allowed to get. It's important to mention that fewer levels in the hierarchy mean simpler and quicker distributions of user and group profiles. Each new layer will increase distribution times.

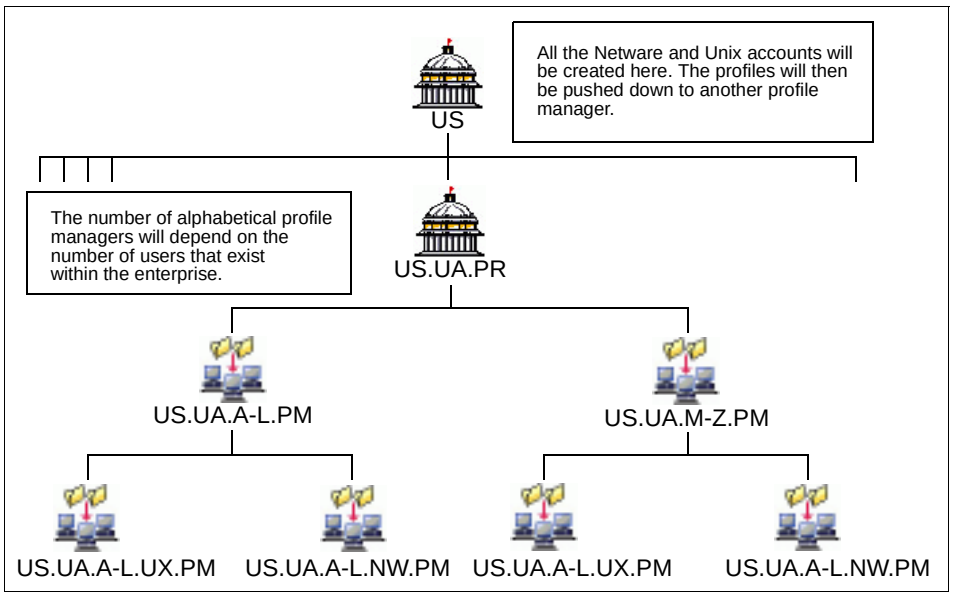


Figure 9. Top-Level User Administration Policy Region

In Figure 9, we show a breakdown of the user administration region. Being at the top level and existing outside the context of the regional policy regions

allows us to control the administration of user accounts for the entire enterprise in the centralized manner we desire. You can see that the user administration policy region (US.UA.PR) contains two profile managers. These break up all user accounts first into alphabetical groupings, and at the next level, into system disciplines.

The number of alphabetical breakdown profile managers will be determined by the lead user administrator. There is a long-standing recommendation to keep the number of users in a single profile below 500. There is a potential for performance problems in Tivoli's management of the profile, and in any case, it eases administration to break it up into more manageable chunks. The 500 limit may at one time have been a physical coded restriction. There are no longer any coded restrictions but larger numbers can still result in resource-related problems.

See also, the discussion on profile design in "User Administration Profiles" on page 56.

Each time a distribution takes place, the distributed profile is merged with the previous version of the profile already present on the target, and ultimately, merged with the system files. This merge must take place whether there are 5 or 500 changes in the profiles. This merging process also leads to the suggestion that you avoid too many levels of subscription from one profile manager to another. The typical recommendation is to limit the number of profile managers in a subscription chain to no more than three.

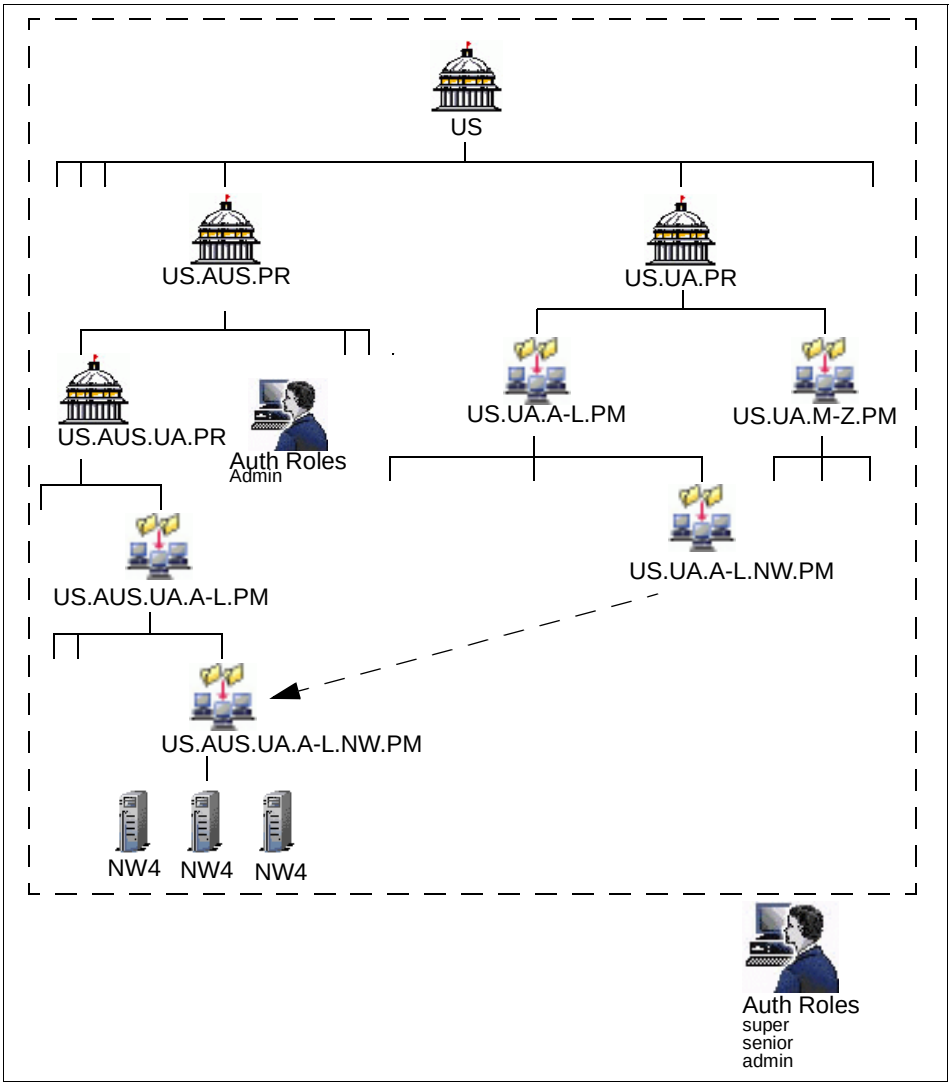


Figure 10. Administration Role and Site-Specific Policy Region Relationship

The configuration in Figure 10 allows user profiles, modified by an administrator in the US.UA.PR policy region, to be pushed down to the site-specific profile managers (indicated by the dotted-line arrow). You can see how the user US.UA.A-L.PM and US.UA.M-Z.PM profile managers under the US.AU.PR policy region do not have endpoints subscribing to them. Instead, they have as subscribers the site-specific profile managers, who, in turn, have endpoints as subscribers. In this example, all NetWare and UNIX

server endpoints will subscribe directly to the alphabetical profile manager in their own user administration profile manager under the Austin policy region.

3.2.2 Distributed Administration

In a distributed administration design, you use localized system administration. This means having administrators at various operational sites responsible for administering the user resources. As with the centralized administration design, described in 3.2.1, “Centralized Administration” on page 32, the distributed model of administration can apply independently of whether the management systems themselves are centralized or widely distributed.

3.2.2.1 Physical Layout

Figure 11 shows the distributed physical layout. This is similar to Figure 4 on page 34, with the important difference being, that each locale has its own administrator - here represented as being split into separate TMRs. Note that if the administrator in TMR A needed to modify profiles after distribution to the other TMRs, that administrator would need the relevant roles in those TMRs too.

Each administrator has all the relevant roles for managing their own TMR.

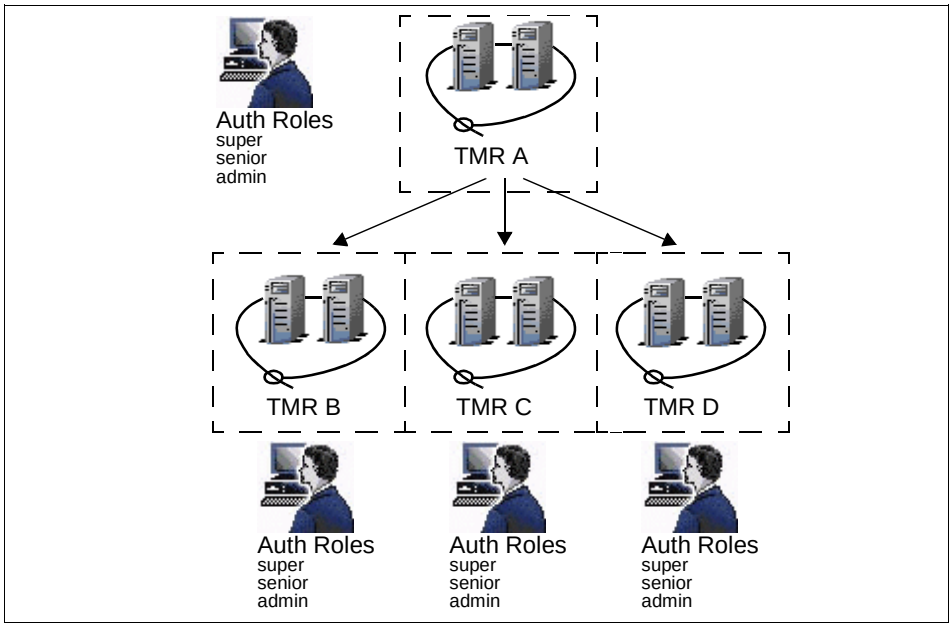


Figure 11. Distributed Administrator Roles - Distributed Physical Layout

If you have distributed TMRs, it may be better to have administrators for each of them. This helps in areas such as fault resiliency. If TMR A server went down (for example, a power outage or communication failure), TMR B, TMR C, and TMR D can continue to be managed as normal. Local administrators continue to manage the resources in TMR B, TMR C, and TMR D.

Note

Multiple TMRs by themselves do not provide system redundancy for down TMR servers. If a TMR server fails, management in that TMR ceases until such time as the server can be reinstated. For TMR server high availability suggestions, see the *Implementing TME 10 in High Availability Environments*, SG24-2032, redbook.

Figure 12 could either be the next level of administration for each TMR described below, or it could be the model for distributed administration in a physically centralized environment. One lead administrator manages the account information for the whole TMR, but individual administrators may manage certain pieces of account information on their own account management server(s).

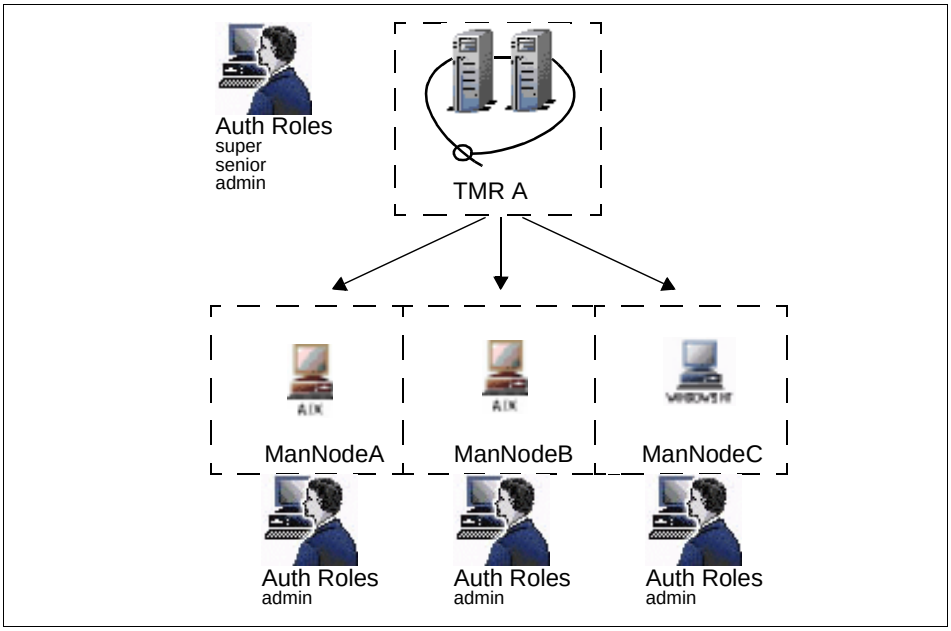


Figure 12. Distributed Administrator Roles - Centralized Physical Layout

3.2.2.2 Policy Regions and Profile Managers

Continuing with the example from 3.2.1.2, “Policy Regions and Profile Managers” we will describe how the distributed administration model would apply to the same scenario.

In Figure 13, we don’t have a top level user administration policy region. We have defined only the policy regions for the regional TMRs.

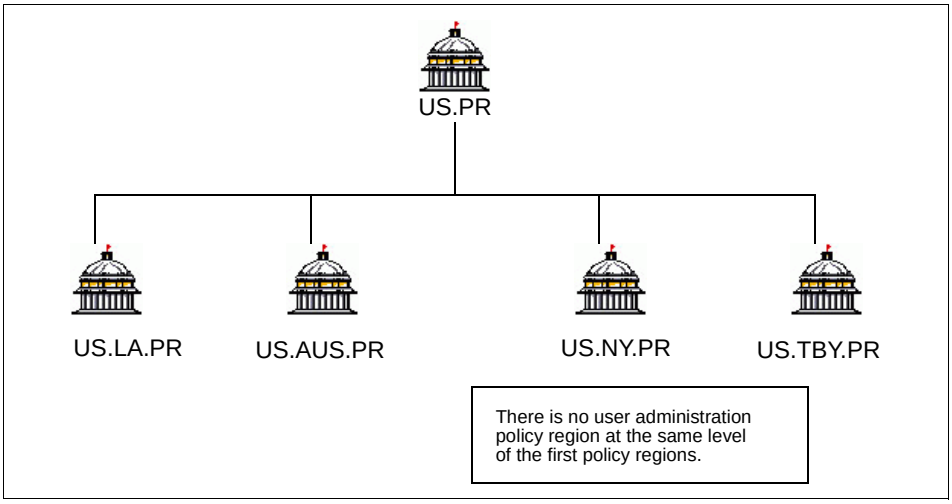


Figure 13. Distributed Administration - Top Level Policy Region Design

Since we don’t have a user administration top level region, the user administration profiles are distributed among each policy region, and each policy region has its own user administration profile managers.

Figure 14 shows how you can distribute your administration over a centralized layout. In our example, we use an overall administrator with powerful roles and distribute the administration of day-to-day operations with junior administrators working with the managed nodes.

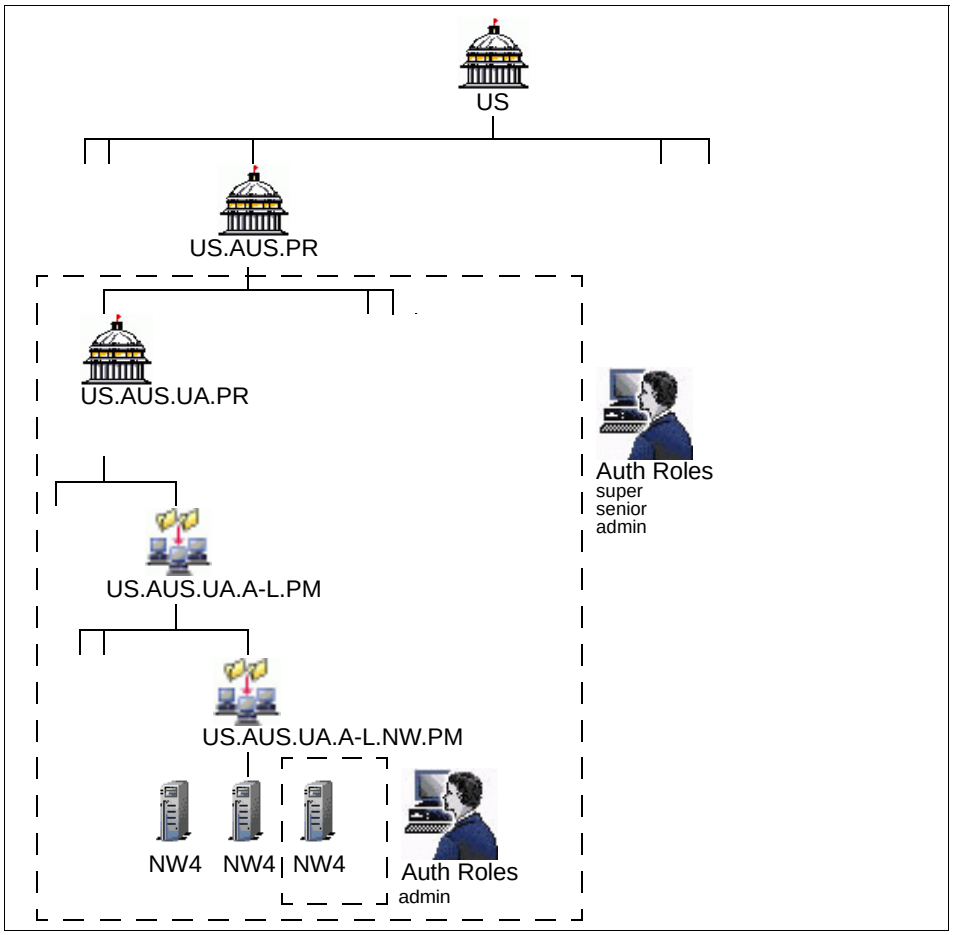


Figure 14. Site-Specific Policy Region Design

3.2.3 Tivoli Object Naming Considerations

This section is designed to help define additional policies and standards in preparation for the implementation of a finished design.

Components managed with Tivoli are represented by objects in the Tivoli management database. A master copy of this database is held on the TMR server, and each managed node maintains its own database of data relevant to its own operation. If a Tivoli administrator creates a new object of a supported Tivoli resource class, for example, a Profile or a Managed Node, the TMR server automatically assigns a new object identifier (OID) to the

object. This identifier is a unique number, even if all TMRs of the company are connected with one- or two-way connections.

If Tivoli manages everything through uniquely numbered objects, you might think that names given to those objects are not all that significant. An administrator will work with a resource name of a Tivoli object, either from a command line or the management desktop. The resource name is a *label attribute* of the resource class to which the object belongs. A unique resource name is the basis on which the resources of different TMRs are distinguished. Tivoli, for the most part, will enforce TMR-specific names, but it is allowable, for example, to have a profile and a profile manager with the same name. Distinguishing between the two, particularly in error logs or when using the command line, can become tedious. So, what we recommend is a naming convention that not only ensures that object names are unique, but also adds some indication of the type of object in the name.

The standard naming convention for the resource label is very flexible in Tivoli. The string can be longer than any administrator would wish to make it and can include almost any special character, excluding the pound sign (#). Note that the names are case sensitive.

The best design for a naming convention can depend very much on the implementation environment. We suggest the following guidelines for defining Tivoli resource names. The first list is a set of general rules:

- Keep the names as short as possible.
- An object's name should immediately identify the type of object.
- Use all same-case letters.
- Avoid special characters except one basic sign, such as dot (.) or dash (-) to separate parts of a name.

In order to more easily distinguish resource names, we suggest adding some organizational information to a description of the function of the resource. This leads to the following naming structure suggestions:

- Include two letters (either leading or trailing the name) describing the resource type.

  Every object belongs to a specific resource class. We append a code representing the class to every resource name. Some suggested codes for the resource classes are listed for user administration-related resources in the sections following this one. An example might be up for user profile. This would give the name the form of up-objectname or objectname-up, which is instantly recognizable as a user profile.

- Three letters describe the scope of the resource.

  This part of the name can be used to show which TMR the resource belongs to. We need this distinction when using different management policies in each TMR. This allows us, for example, to define a system user profile for several countries where the rest of the resource name is equal:

  - `up-mxx-system` is the user profile for the Mexico TMR.
  - `up-uxx-system` is the user profile for the USA TMR.

  The scope of the resource is built of a one-letter code applicable to the geography containing a TMR (such as a country) and two letters for the specific location or division.

  If a resource applies to all TMRs, we can use the string `xxx` as scope or omit it altogether - as long as we remain consistent. Similarly, if the resource has been defined for all TMRs of a country, we use the code of the country together with the string `xx` as in the above examples.

- Two letters specify the operating system, if necessary.

  If a management resource contains platform specific information, we may have to add the code of the operating system as part of the resource name. This procedure is very helpful for the administrators to choose the correct targets for the distribution of profiles. In addition, this allows us, for example, to define a user profile for several operating systems, where the rest of the resource name is equal:

  - `up-xxx-ai-system` is the user profile for AIX that applies to all regions.
  - `up-xxx-nt-system` is the user profile for Windows NT that applies to all regions.

  If the resource is platform independent, we can use the string `xx`, instead of the code for a operating system, or omit such markings altogether. Again, we must remain consistent - if no marking is used, and the resource is platform specific, this could introduce problems.

  We suggest the definition of a table for use by administrators. This would contain the codes for all supported platforms. This table can be used to validate resource names or could potentially be added as validation policy in Tivoli.

- One or more letters describe the function or task of the resource.

  The description about the management function or task of the resource should be as short as possible, while maintaining an obvious meaning for every administrator.

Avoid the use of superfluous words, for example, `sp-xxx-ai-security system policy for aix`. The information resource type and platform are already included in first and third part of the name.

This naming convention is just an example. You can add additional information to the resource names, for example, the name of the application that is managed with the Tivoli object.

The next sections give a few more examples of naming convention ideas for different Tivoli components.

3.2.3.1 Tivoli Management Region Names

Every TMR server can either be accessed with the region number that is automatically assigned during the installation or with the TMR name. You can think of this name as an alias for the server.

In many cases, it may be useful to incorporate the country code, and certainly, the location or division code where the server resides. For example, `mme` for a TMR server in Mexico City, Mexico. Examples of codes are defined in the following tables.

1. Country codes, where the company has locations or divisions:

Table 9. Valid Country Codes for the Naming Convention

Country Code	Full Country Name
a	Australia
g	Germany
m	Mexico
u	United States of America

2. Code for locations or divisions in the company:

Table 10. Valid Location Codes for the Naming Convention

Location Code	Full Name of the Location or Division
au	Austin
sy	Sydney
mu	Munich
me	Mexico City

This approach for standardizing the TMR name has an advantage when creating local resources. The command `wtmrname` returns the name of the TMR and can be used in a shell script when assigning a resource name to a new Tivoli object. The following is an example of creating a new group profile from a standard profile, where the TMR name is incorporated into the name of the new group profile - note that this can be run on any TMR without modification, and it will correctly use the name of the local TMR.

```
wcrtprf -c @GroupProfile:gp-xxx-xx-standard \
   @ProfileManager:pm-'wtmrname'-xx-system \
   GroupProfile gp-'wtmrname'-xx-system
```

3.2.3.2 Tivoli Management Framework Name Codes

The naming convention cannot be applied for every Tivoli resource. There are some types that use the same fixed name from the TMR server. The following resources have names that are equal in every management region:

- The Desktop resource type (TME Desktop).
- Administrators is the name of the type AdministratorCollection.
- Notices is the name of the type BulletinBoard.
- Scheduler is the name of the type Scheduler.

Because we want to add the type in the resource name, we have to define a code for the framework resources. Table 11 can also be used to validate the names of resources:

Table 11. Framework Resource Naming Codes

Resource Class	Code	Example
PolicyRegion	pr	pr-gmu-xx-users
PolicySubregion	pr	pr-gmu-xx-users development
ProfileManager	pm	pm-gmu-xx-users development
TaskLibrary	tl	tl-xxx-ai-users
Task	ta	ta-xxx-ai-users check security
Job	jo	jo-gxx-ai-users check security

There are several resource types for which we do not suggest applying the naming convention for Tivoli objects:

- Administrator

  It is more convenient to use the full user name for the resource type Administrator. One possible exception might be a situation where multiple

administrators use the same Tivoli administrator ID. For example, all those charged with adding users may use a name, such as `ad-gmu-xx-adduser`. Shared administrator IDs are discouraged in Tivoli for accountability reasons.

- GenericCollection

  A generic collection is normally used by administrators to group resources on their own desktop. That name is only relevant for the owner of the generic collection, the administrator.

- Endpoints

  It makes more sense to assign the TCP/IP host name or NetBIOS computer name to the Tivoli resource name.

3.2.3.3 Tivoli User Administration and Security Management

Table 12 contains the codes of resource types from Tivoli User Administration and Tivoli Security Management to be used as part of the naming convention for Tivoli objects.

Table 12. User Administration Resource Naming Codes

Resource Class	Code	Example
UserProfile	up	up-gmu-ai-development
GroupProfile	gp	gp-gmu-ai-development
SecurityProfile	sp	sp-xxx-ai-system

In Figure 15, we will apply all the naming conventions above to an example based on Figure 8 on page 37.

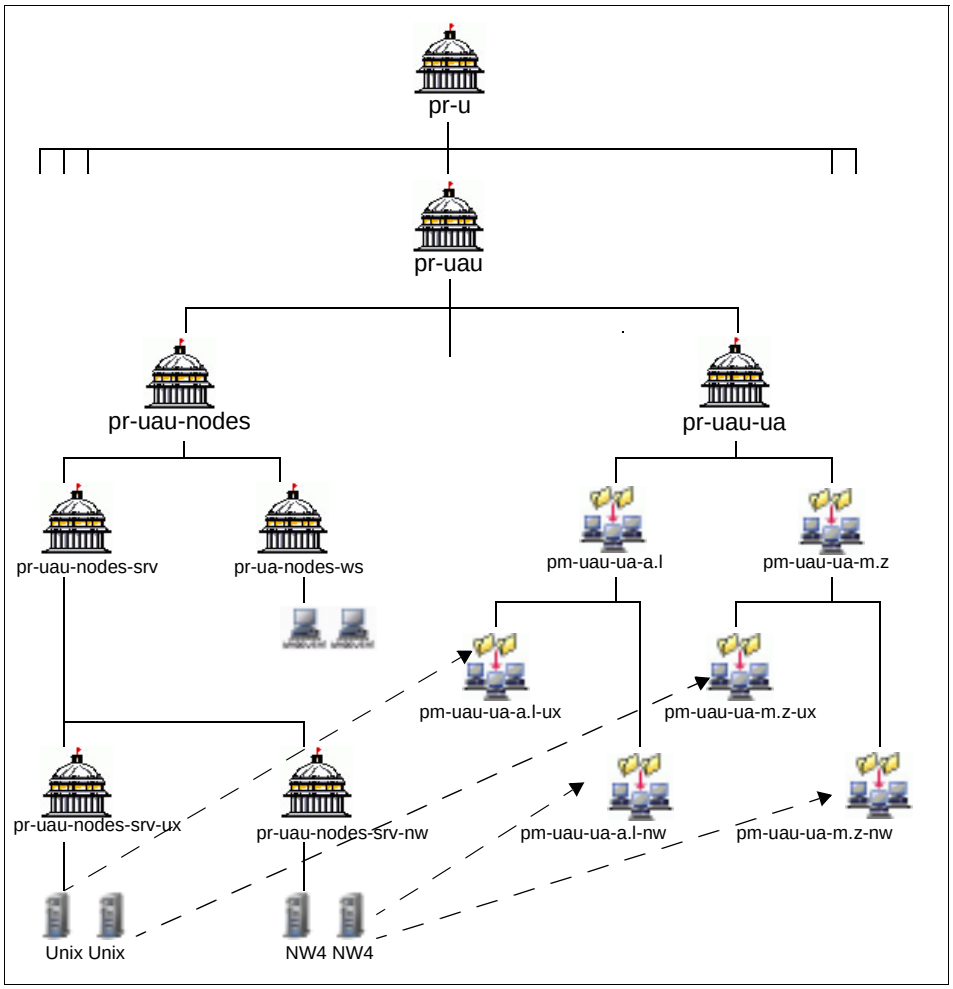


Figure 15. Alternative Naming Convention

The naming convention topic is also discussed in the *Tivoli Security Management Design Guide* (IBM order number SG24 5101).

3.3 Physical Installation Considerations

Before starting any Tivoli implementation, we strongly recommend that you read and understand the *Tivoli Management Framework Planning and Installation Guide,* and that you read the release notes for all products you will install. The *Tivoli Management Framework Planning and Installation Guide*

provides a comprehensive description of the Tivoli Management Framework and guidance in planning your Tivoli installation.

The remainder of this section highlights of the most important hardware considerations for the Tivoli Management Framework and Tivoli User Administration.

3.3.1 Planning a Framework Installation

The *Tivoli Management Framework Planning and Installation Guide* will be your main source of reference for framework installation planning. You may also wish to become familiar with other redbooks, such as *Mass Installation of TME 10 Endpoints*, SG24-5109.

3.3.1.1 TMR Server Considerations

The machine used as a TMR server is the most critical part of your Tivoli installation. It should be a high-performance machine, very stable, and with as few as possible non-Tivoli applications running on it. The best choice is a dedicated machine. The following table describes some acceptable minimum configurations for a TMR server.

Table 13. Typical Configurations for a TMR Server

System	RAM	Swap Space
Data General AViiON Series 530	64 MB	128 MB
HP 9000/735	64 MB	128 MB
IBM RS/6000	64 MB	128 MB
Intel 486 or Pentium running Windows NT 3.51	48 MB	128 MB
Intel 486 or Pentium running UnixWare 2.0	48 MB	96 MB
Motorola 88000 Series	64 MB	128 MB
NCR 3400/3500 Series	64 MB	128 MB
SPARC 10 or SPARC 20 running SunOS4.1x	48 MB	96 MB
SPARC 10 or SPARC 20 running Solaris 2.3	64 MB	128 MB

The type of machine to choose as a TMR server depends on the complexity and the size of the environment to manage. If your Tivoli installation has only a small number of machines to control, an entry level machine is acceptable. If you plan to manage an environment of (for example) hundreds or thousands of machines, a higher level machine is required. With the 3.6 release, Tivoli performed a great deal more capacity and scalability testing.

We can expect some documentation to emerge from Tivoli based on the results of that testing.

Remember that a single server can manage up to 200 Tivoli clients and many thousands of Tivoli Management Agents. If your environment is larger, you will need to consider additional TMR servers and connections between them.

Tivoli recommends limiting the number of Tivoli clients (Managed Nodes) to 200. No such limitation exists for PC Managed Nodes or Tivoli Management Agents. In installations at version 3.6 or above, the more efficient way to manage endpoints is to use the Tivoli Management Agent and reserve the use of managed nodes to those application that still require a managed node or for use as gateways for Tivoli Management Agent endpoints.

3.3.1.2 Tivoli Management Stations

A Tivoli management station is a TME client machine from which an administrator can perform operations by using the Tivoli desktop. This machine must be able to run Win32 based or Motif applications and must have good networking performance (to provide fast, reliable communication with the TMR server). Even for these machines, there are minimum architectural requirements. These are listed in the following table:

Table 14. Typical Configurations for Tivoli Management Stations

System	RAM	Swap Space
Data General AViiON Series 530	48 MB	96 MB
HP 9000/735	48 MB	96 MB
IBM RS/6000	48 MB	96 MB
Intel 486 or Pentium running Windows NT 3.51	32 MB	64 MB
Intel 486 or Pentium running UnixWare 2.0	32 MB	64 MB
Motorola 88000 Series	48 MB	96 MB
NCR 3400/3500 Series	48 MB	96 MB
SPARC 10 or SPARC 20 running SunOS4.1x	48 MB	96 MB
SPARC 10 or SPARC 20 running Solaris 2.3	48 MB	96 MB

3.3.1.3 Tivoli Managed Nodes

The Tivoli client (UNIX or NT) should have enough disk space to support the installation of the Tivoli client part and a configuration of 32 MB of RAM and 64 MB of swap space. Tivoli does not impose a heavy load on the client machines when idle, as far as management operations are concerned.

However, the load increases when Tivoli operations are performed on the client. It's common practice to schedule the most demanding Tivoli operations during hours in which the machine is not busy. For example, you may want to schedule the distribution of a new user profile made up of 100 users on your Windows NT server to occur during the night, when the fewest users are logged on to the server. To perform the installation of a Tivoli client, you have to be able to access the client machine as Windows NT Administrator or UNIX root.

3.3.1.4 PC Managed Nodes

The minimum requirements for a Tivoli PC managed node are listed in the following table:

Table 15. Configurations for Tivoli PC Managed Nodes

Supported Platforms	Minimum RAM Requirements
NetWare	8 MB
OS/2	8 MB
Windows (WFW, W3.x, W95)	2 MB
Windows NT	8 MB

3.3.1.5 Tivoli Management Agent

The Tivoli Management Agent is available for UNIX systems, PCs running Windows 3.11, Windows 95, and Windows NT, with support coming for OS/390, OS/2, and others. Check the release notes for the software you have (from 3.6 onwards) to determine the correct hardware requirements.

3.3.1.6 Communications Considerations

The basic requirements for all the Tivoli operations is a bidirectional TCP/IP connection. If you have a line slower than 14.4 KB (for example a WAN line between two remote sites), Tivoli clients can continue to communicate with one another, but you will not be able to remotely install Tivoli clients. In this case, the installation can only be performed locally.

You must also check that every client is able to perform the mapping between IP addresses and host names (reverse mapping). Before installing a client, this mapping must be working on the client, as well as the server, in order to establish the initial connection.

To determine if reverse mapping is available on a machine, you can execute the following command:

`nslookup nnn.nnn.nnn.nnn` on a Unix or Windows NT system

or

`ping -a nnn.nnn.nnn.nnn` on a Windows NT system

where `nnn.nnn.nnn.nnn` is the IP address of the system you want to check. If this command returns you the name of the host that corresponds to the IP address, your Domain Name Service is configured for reverse mapping. If you don't get the host name, you can do the following:

- Add the IP address to host name maps for the domain name server (DNS)

or

- Use the LMHOSTS facility on Windows NT, or add the hosts to the /etc/hosts file, and use the /etc/hosts file as a DNS fall-back.

In a Tivoli environment, IPX/SPX communications are supported between a NetWare server, defined as a PC Managed Node and NetWare clients (PC Endpoints). However, only TCP/IP is supported between a TMR server and a NetWare PC Managed Node. You must ensure, therefore, that TCP/IP is properly configured on your NetWare server.

3.3.2 Planning a Tivoli User Administration Installation

After planning your Tivoli Management Framework installation, you can plan the installation for Tivoli User Administration.

Depending on the version to be installed, there may be both a product install and an patch upgrade, such as installing version 3.0 and then adding 3.1.2 or from version 3.1 and adding 3.1.2. Version 3.6 is a stand-alone installation, but you will need to watch for patches that may be required to support additional endpoint types, such as AS/400, OS/390 Security Server, and OS/2.

Note

This document describes a typical Tivoli User Administration installation but not patch installations. You should always review any product release notes for the latest information about installation prerequisites and other

Refer to the Release Notes for the version you are installing, but Tivoli User Administration now covers a wide variety of platforms, including:

- Various UNIX versions (AIX, HP-UX, Sunsoft Solaris, SunOS)

- Windows NT
- NetWare
- Lotus Notes Version 4.1
- Domino/Notes 4.5
- OS/390 Security Server
- OS/400
- Lightweight Directory Access Protocol (LDAP)

Note

The Tivoli Management Framework is not supported on Windows NT multi-user add-ons such as WinDD or Citrix Winframe or on NT machines running beta releases of the Windows 95 shell or New shell.

3.4 Additional Design Considerations

This section includes further notes to complete the discussion on the design of a Tivoli User Administration solution. Presented here are examples of heuristic knowledge, usually obtained through repeated experience to implementations.

In this section, we will give you some useful rules of thumb - guidelines and suggestions on the best practices.

3.4.1 Policy Regions

If you are trying to implement a distributed user administration environment (see 3.2.2, “Distributed Administration” on page 41), you should be assigning Tivoli roles to Administrators per policy region and restricting the TMR-wide roles assigned to limit access to policy regions intended for management by other administrators.

If you have a centralized environment (see 3.2.1, “Centralized Administration” on page 32), you should group user profiles into a separate policy region dedicated to Tivoli User Administration. This can be significant in helping to decrease desktop clutter and make user administration more efficient.

3.4.2 Managed Resources

The installation instructions remind you about adding managed resources to a policy region, but it is still easy to forget this step. To manage users and groups within a policy region, you must add UserProfile and GroupProfile (as well as ProfileManager) as managed resources in that policy region. You can use the `wsetpr` command or use the Set Managed Resources dialog in the

GUI from the Policy Region's context menu or from the Properties menu of an open region window - see Figure 16.

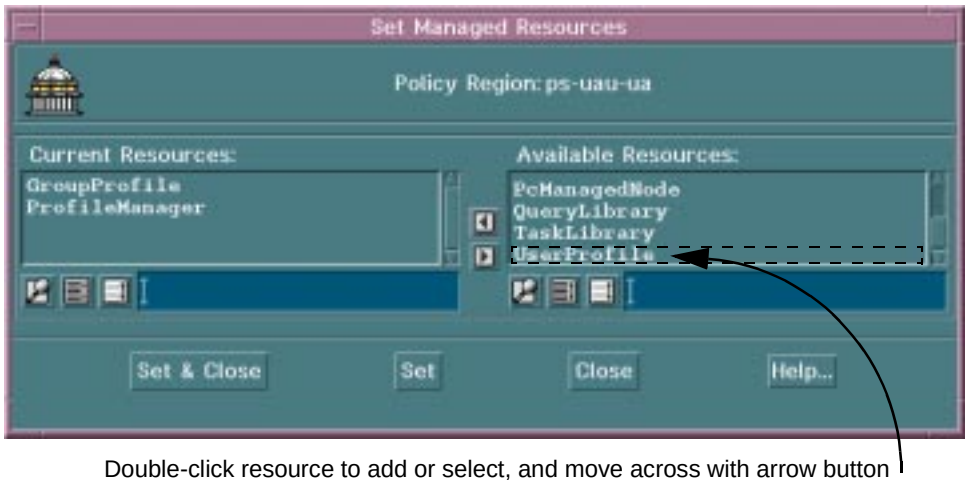


Figure 16. Adding a Profile and Profile Manager as Managed Resources

3.4.3 User Administration Profiles

It is a common recommendation to restrict the number of users defined in a single profile. There are no hard and fast numbers, but typically, the number given by Tivoli User Administration implementation experts is around 500'.

The reasons for the restrictions come from the resources required to manage large amounts of user account data. A distribution of a user profile sends the profile to the endpoint. Once there, the endpoint's existing user data is read (from /etc/passwd or wherever). The user data is compared and merged, as appropriate with the profile data, and the resulting information is written back to the system user data files. If the profile that is distributed contains 5000 users, but the /etc/passwd file contains only four users, the entire profile is still delivered to the endpoint, and the merge must still check the entire profile of 5000 records.

Breaking up the user records into smaller numbers reduces the overhead associated with profile transactions. Large profiles can also affect the performance of the GUI. Tivoli development continues to review the management of users and profiles to determine the best trade-off between functionality and performance.

If you have an environment with more than 500 users, it is normally not too problematic to break up the list into smaller units. Ideas for doing this include

using UNIX UID ranges, alphabetical user name ranges, users locations, users departments, and so on. Some examples of doing this are in Figure 17. Note that during population of user profiles, this division must have been decided upon ahead of time. The split lists of users can then be used as input to the `wpopusrs` command to ensure only those users in the right range are populated into the profile. Note also, that even though the framework populate command, `wpopulate`, can be used to populate user profiles, it has no such facility to just populate given user names from a list in a file.

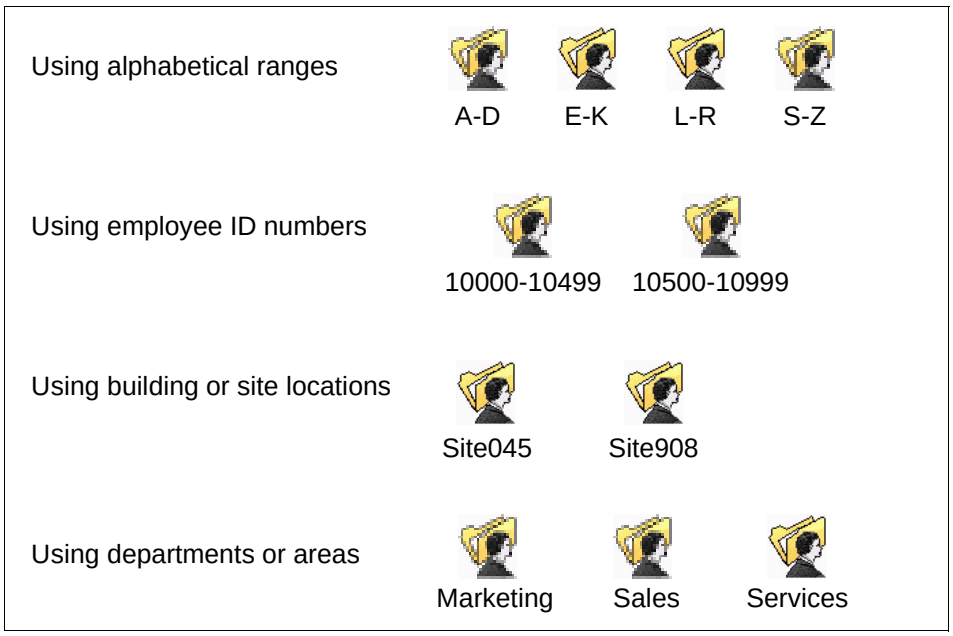


Figure 17. Defining Manageable User Population Chunks

3.4.4 Administrative Roles

If you have a centralized user administration environment (see 3.2.1, “Centralized Administration” on page 32), it stands to reason to keep the most powerful Tivoli roles (Admin and Senior) to the central administrators and delegate user roles to other administrators. The other administrators would have sufficient authority to run tasks designed by more powerful administrators but not to modify those tasks or perform more sensitive operations.

An example of delegating roles would be to consider a help desk operation. Delegating authority to help desk and operations personnel for user control

and other standard tasks enables faster problem resolution, while maintaining control. A common task library could be created by senior user administrators and used by the help desk operators when required.

This approach provides a formal, secure mechanism for automating and defining solutions to routine problems, such as reactivating a locked out user, by taking advantage of the Tivoli delegated administration model. Note that this delegation of authority can apply to the centralized or distributed model, although in the distributed model you are generally using more powerful administrators in each location.

If you have a distributed user administration environment (see 3.2.2, “Distributed Administration” on page 41), distribute roles (Admin and Senior) to administrators in charge of managing their own policy regions, and optionally, avoid granting the same roles for those administrators for regions outside of their immediate control. This decision depends on the policy regarding the interaction of different administrators.

3.4.5 Implementation

Many environments have users managed and added in a piecemeal fashion. Some systems implement a naming convention, for example, and other parts of the organization don’t use it. If you have an environment without well defined standards and policies, it is advisable to work through the following steps before implementing Tivoli User Administration:

- Change user login names, so you have only unique login names across the enterprise.
- Likewise, change UNIX UIDs, so you have only unique UIDs across the enterprise.

  (When changing UIDs, be sure to also change ownership of data)
- Implementing NIS used to make better sense, rather than making every UNIX machine a Tivoli endpoint, and it would avoid the need to distribute the password and group files to potentially hundreds of managed nodes. However, with the introduction of the Tivoli Management Agent, you can minimize the overhead associated with managing large number of UNIX systems and maintain the consistency of management through a single interface.
- When allocating time and resources for implementation and deciding on the order of implementation, it is advisable to identify how complex each platform will be to implement. Typically, UNIX will be the easiest to start managing with Tivoli User Administration. Windows NT is likely to be more

complicated, especially if you will be managing multiple domains, while Novell NetWare may lie somewhere between the two, for example.

Chapter 4. Implementation

Having finished the design, we can now look at how to implement it. We need to consider factors, such as the number of users we need to store in the database, the best way to perform this implementation (phases, duration), how long it will take, what kind of people do we need to perform the implementation, and so on. We can also take a look at anticipating problems that could occur in the implementation.

4.1 Implementation Activities

This section details some common activities that you may get involved in during an implementation.

4.1.1 Cloning User Profiles

In many Tivoli User Administration implementations, there will be some modification to the user profile in use (see “Modifying Default and Validation Policy” on page 68, for example). That modification will need to be applied to any new profiles that are created. Typically, this means that in Tivoli User Administration, new user profiles are not created but cloned from an existing profile used as a template. The profile that is used when you select to create a new profile (for example, using the profile manager edit menu option **Profiles --> Create** in the GUI) is called TivoliDefaultUserProfile. Cloning involves copying an existing profile, not using the system default or the create options.

Note

When using a customized user profile as a template, it is advisable to use a new profile rather than edit the system default user profile (TivoliDefaultUserProfile). There may be occasions in the future when the default profile is required. It is also possible that patches to Tivoli User Administration could modify the system default user profile.

To create a clone of an existing profile, use the profile manager edit menu option **Profiles --> Clone** or use the `wcrtprf` command with the `-c sourceprofilename` option.

4.1.2 Modifying Tivoli User Administration With AEF

In many installations, there may be some need to modify the way Tivoli User Administration works in order to make it fit in better to the specific

environment in which it is implemented. Usually, this stems from a requirement to manage additional attributes to those provided by default within the product. Tivoli provides the capability to perform such changes through the Application Extension Facility (AEF). AEF can be used on other products, but it is used most when implementing Tivoli User Administration.

This section provides an example by extending the Windows NT management capabilities of Tivoli User Administration by adding a Windows NT home directory management attribute. Most of the add-on products that integrate with Tivoli User Administration make use of AEF to provide that integration. Tivoli is in the process of producing a Tivoli User Administration tool-kit to help ease the creation of user administration extensions. Contact your Tivoli representative for further details about this kit.

4.1.2.1 Managing Windows NT Home Directories

Tivoli User Administration does not create home directories for Windows NT users when the profile is distributed, unless you customize the application to do it. In this section, we show an example of customizing with AEF that allows you to automatically create the corresponding user home directory on a specific server.

AEF, provided with the Tivoli Management Framework, allows you to customize Tivoli applications and is frequently used to extend Tivoli User Administration. AEF might not be installed on your TMR server, but installing AEF is easy. As with any other Tivoli application, you just need to go to the Tivoli desktop and select **Install**.

Tivoli User Administration divides management attributes or properties into categories, such as UNIX or NT, and subcategories, such as UNIX Login, NT Password, and so on. When you are managing Windows NT users, you can select the NT category, and only Windows NT subcategories are listed in the GUI. You then select the subcategory of interest, and the properties associated with that subcategory are displayed in the panel. We can use AEF to add properties as well as actions that will take place on a distribution, and Tivoli User Administration includes commands to enable us to add categories and subcategories.

AEF and the Tivoli User Administration commands are used in this example to add to the user administration GUI a subcategory that we will call NT_Server in the NT Category scrolling list. This subcategory allows us to specify the name of the Windows NT server on which we want to create the home directory for a user.

Then, we will add an action to the user profile. This action will be executed right after the profile distribution and will actually create the home directory. Note that this action is for a Windows NT managed node and would need to be different for Windows NT with the Tivoli Management Agent.

You will find below all the steps necessary to create that subcategory and add the home directory creation action:

1. Add a new property to the user profile:

   ```
   waddprop @UserProfile:<profile_name> "NT_Server" ""
   ```

   In our case, we entered the following:

   ```
   waddprop @UserProfile:NT_Users "NT_Server" ""
   ```

   The NT_Server attribute indicates the Windows NT machine on which the user home directory will be created.

   You can check that the new attribute has been properly added in the User Profile Properties by clicking **Edit** -> **Default Policies** to get the window shown in Figure 18.

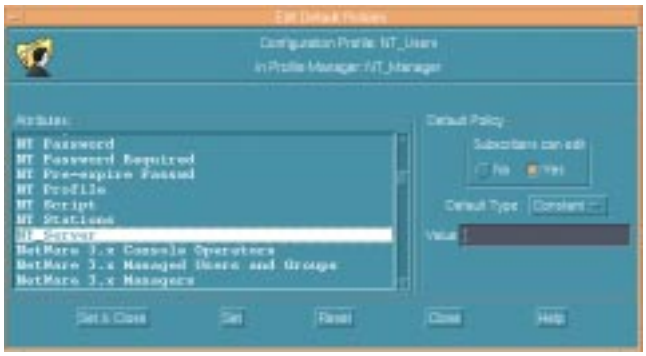


Figure 18. AEF NT_Server Default Policy

   You will find that the `waddprop` operation has also defined a default policy constant for the NT_Server attribute. This new property is also added as a column into the user profile properties window.

2. Create the NT_UserInfo subcategory in the NT category by issuing the following command:

   ```
   wcrtusrsubcat -m "NT_Server" -c NT NT_UserInfo
   ```

   NT_UserInfo is the name of the subcategory dialog that will be displayed when selecting NT_Server in the NT Category scrolling list. Figure 19 shows the new NT_Server entry in the NT Category scrolling list.

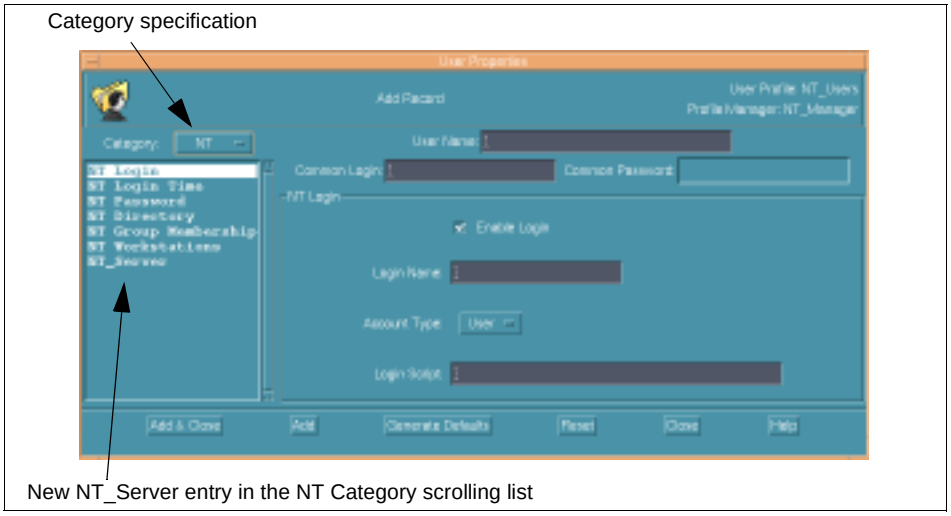


Figure 19. AEF NT_Server New Option

3. Create the dialog specification language (dsl) source file /tmp/nt_server.dsl. This file should look like the following:

```
Partial Dialog
{
        Gadgets
        {
            Group
            {
                Attributes
                {
                    Border = YES;
                    Name = NT_UserInfo;
                    Title = "NT UserInfo";
                    TitlePos = TOP;
                    Visible = NO;
                    GridHorizontal = 0;
                    GridVertical = 0;
                    ChildColumnAlignment = STRETCH;
                    ChildRowAlignment = STRETCH;
                }

                Gadgets
                {
                    Group
                    {
                        Attributes
                        {
                            Layout = VERTICAL;
                            Name = NT_UserInfoContainer;
                            ChildColumnAlignment = LEFT;
                            ChildRowAlignment = STRETCH;
                        }

                        Gadgets
                        {
                            Text
                            {
                                Name = NT_Server;
                                Title = "Server for Home Directory";
                                TitlePos = TOP;
                                ChildColumnAlignment = LEFT;
                                ChildRowAlignment = LEFT;
                            }
                        }
                    }
                }
            }
```

Note the entry `Name = NT_UserInfo`. This matches the name we used when adding the subcategory.

4. Put the dialog in all user profiles by entering the following command:

   ```
   dsl /tmp/nt.server.dsl | wputdialog -r UserGui NT_UserInfo
   ```

5. Create a shell script called mk_nt_home_dir.sh. This shell script will be executed when the profile is distributed. The shell script should look like the following:

```
#!/bin/sh

if [ $# != "2" ]
then
        echo "Usage: mk_nt_home_dir < hostname > < User > ">>/tmp/debug
        exit 1
fi

DIRMODE=0777
DIROWNER=Administrator
DIRGROUP=Administrators
HOST=$1
DIRPATH="c:/users/"$2
MNO=‘wlookup -r ManagedNode $HOST‘

idlcall -T top $MNO \
"TMF_ManagedNode::Managed_Node::make_directory" \
"\"$DIRPATH\" { $DIRMODE -1 \"$DIROWNER\" -1 \"$DIRGROUP\" -1 } 1" >> /tmp/debug
2> &1
```

Note that in this script, we create the home directory under C:\users. And the home directory will be C:\users\<login_name>. You can, of course, customize this shell script for your own needs.

6. Add an action to the user profile when the profile is distributed. That action will be executed on the TMR server itself. Enter the following command:

   ```
   waddaction -c -a @UserProfile:<profile_name> mk_nt_home_dir \
   args=’$NT_Server’,’$nt_logon’ < mk_nt_home_dir.sh
   ```

   In our example, enter:

   ```
   waddaction -c -a @UserProfile:NT_Users mk_nt_home_dir \
   args=’$NT_Server’,’$nt_logon’ < mk_nt_home_dir.sh
   ```

   We can retrieve information from the user properties as arguments to our action script. The Tivoli User Administration property `$nt_logon` corresponds to the login name of the user. Our custom property we added, `$NT_Server`, is the name of the Windows NT server where the home directory will be created. If you want to know what the names are of the properties in a panel, you will need to retrieve the dsl code.

If you want to retrieve the code corresponding to the NT_Server panel, you need to perform the following steps:

1. List all the dialogs associated with the user interface by issuing:

   ```
   wlsdialog -r UserGui
   ```

   You will get the output shown in Figure 20.

```
dialog name (customization status)

AddEditUser
DelUserConfirm
EmptyPage
IdentificationGroup
NDSBrowser
NTDirectoryGroup
NTGroupMemGroup
NTLoginGroup
NTLoginTimeGroup
NTPasswordGroup
NTWorkstationsGroup
NW3ManUserGroup
NW3ManagersGroup
NW3MgmtRightsGroup
NWAddrRest
NwDirectoryGroup
NWForEmail
NWForMailGroup
NWGroNWLoginGroup
NWLoginTimeGroup
NwMailGroup
NwNetworkAddrGroup
NwPasswordGroup
NwSecurityGroup
PostalAddressGroup
SubscribersGroup
UDirectoryGroup
UEmailGroup
ULoginGroup
UPasswordGroup
UserProps
UserPropsIcon
NT_UserInfo (resource-wide customization)
upMemGroup
```

Figure 20. Dialog List

You can see the NT_UserInfo subcategory.

2. Retrieve this dialog by issuing the following command:

   ```
   wgetdialog -r UserGui NT_UserInfo > /tmp/nt_server.d
   ```

3. Reverse compile the file that has been retrieved:

   ```
   rdsl /tmp/nt_server.d > /tmp/nt_server.dsl
   ```

4. You are then able to view the code used for the dialog by issuing the following command (substitute `vi` with your favorite editor/viewer):

   ```
   vi /tmp/nt_server.dsl
   ```

   You will get the output shown in Figure 21.

```
Partial Dialog
{
        Gadgets
        {
            Group
            {
                Attributes
                {
                    Border = YES;
                    Name = NT_UserInfo;
                    Title = "NT UserInfo";
                    TitlePos = TOP;
                    Visible = NO;
                    GridHorizontal = 0;
                    GridVertical = 0;
                    ChildColumnAlignment = STRETCH;
                    ChildRowAlignment = STRETCH;
                }

                Gadgets
                {
                    Group
                    {
                        Attributes
                        {
                            Layout = VERTICAL;
                            Name = NT_UserInfoContainer;
                            ChildColumnAlignment = LEFT;
                            ChildRowAlignment = STRETCH;
                        }

                        Gadgets
                        {
                            Text
                            {
                                Name = NT_Server;
                                Title = "Server for Home Directory";
                                TitlePos = TOP;
                                ChildColumnAlignment = LEFT;
                                ChildRowAlignment = LEFT;
                            }
                        }
                    }
                }
            }
```

Figure 21. Dialog dsl NT_Server Example

4.1.3 Modifying Default and Validation Policy

The use of validation policies and default policies when you are creating user profiles ensure that user accounts adhere to your customer's account creation policy. A number of validation and default policies are already defined and installed with the product. These can be reviewed through the GUI (Edit --> Default or Validation Policies in the user profile), or by using

`wgetpolm`. The policies are all named and described in the *Tivoli User Administration User and Group Management Guide*.

One common activity in the implementation of Tivoli User Administration is the modification of default and validation policies. Sometimes, it may be advisable to disable certain policies to improve GUI performance, and in other cases, it may be necessary to alter or add policies to ensure required operation (such as ensuring UNIX user IDs are in a suitable range within a profile or ensuring the UNIX login name is unique). Here we discuss an example of each of these two techniques.

4.1.3.1 Disabling Default Policies

If you do not manage all the user types supported by a default Tivoli User Administration installation (Windows NT, UNIX or NetWare), you can disable the default policies for the platform you are not using. Disabling un-used policies speeds up the use of the GUI when creating new users. Refer to the `wsetdefpol` command in the *Tivoli User Administration User and Group Management Guide* for a description. The basic format is as follows:

```
wsetdefpol DISABLED platform profile_name
```

Where the policy can be `DISABLED` or `ENABLED`, the platform can be `Unix`, `NT`, or `NW`, and `profile_name` is the name of the profile on which to make the change. If you specify your master profile that you will use to clone to create others (see “Cloning User Profiles” on page 61), all profiles you create by cloning will have the policy disabled.

4.1.3.2 Altering Validation Policies

To ensure correct administration in a large UNIX design, it is important to have unique user names across all user profiles. Validation policy is one way of ensuring this, as it could validate every new user name to ensure its uniqueness.

By default, the standard user administration implementation and validation script for the UNIX login name would allow you to generate a login name that duplicates one in another user profile on the same TMR. However, this script can be modified to check for a duplicate login name by examining the user name registry. More recent versions of Tivoli User Administration may already include this alteration in the script - although it is usually commented out.

The standard script essentially just checks that the login name is all lower case and no longer than eight characters, and it looks like this:

```
#!/bin/sh
#
# Component Name: user_validate_login_name
#
# $Date: 1996/04/20 16:50:48 $
#
# (C) COPYRIGHT TIVOLI Systems, Inc. 1991
# Unpublished Work
# All Rights Reserved
# Licensed Material - Property of TIVOLI Systems, Inc.
#
# Exit Codes:
# E_OK(0)Successful completion
# E_USAGE(1)Illegal option, argument, or parameter
#
MY_NAME=user_validate_login_name
#
# Initialize Exit Codes
#
E_OK=0
E_USAGE=1
#
# Set PATH to known safe value
#
PATH=/bin:/usr/bin:/usr/ucb:$PATH
export PATH
#
# Check that two arguments are specified
#
if [ $# -ne 2 ]
then
echo "Usage: $MY_NAME \"real_name\" login_name"
exit $E_USAGE
else
# Set the variable REAL_NAME and LOGIN_NAME
REAL_NAME=$1
LOGIN_NAME=$2
fi

#
# Make sure the name is between one and eight characters long, is alphanumeric,
# and begins with a letter. All the letters must be lower-case.
#
case "$LOGIN_NAME" in
[!a-z]* | *[!a-z0-9]* | ????????* | '')
echo FALSE
exit $E_OK
esac

echo TRUE

exit $E_OK
```

In order to make this script check for duplicated user names across the TMR, you will need to make some changes to incorporate the `wlsnams` command**.** This command can be used to determine whether or not a user name is in use in the TMR (that is, whether it is reserved in the user name registry). For example, if you type:

```
wlsnams -k @NameDatabase:UserNameDB rhawes
```

the result will be something like:

```
rhawes: "UA-A-L" key=1226016130.1.665_192, "UA-A-L"
key=1226016130.1.665_192
```

This shows that the name is already in use in the profile UA-A-L. If the name was not in the user name registry, the following message would be returned:

```
An instance named “rhawes” of resource “UserNameDB” was not found.
```

To modify the validation policy script, you need to extract it from the user profile where you want to define this validation policy, and redirect the output to a file. The command that you will use for this is:

```
wgetpolm -v @UserProfile:UA-A-L login_name > /tmp/val_login_name
```

This gets the validation policy (-v) from the user profile UA-A-L attribute called login_name. In this case, the policy is a script, which we redirect to the file ‘val_login_name’. Refer to *Tivoli Framework Reference Manual* for more details on the wgetpolm command. The attributes are all listed in the discussion of policies in the *Tivoli User Administration User and Group Management Guide.*

Now you can edit the validation policy script that you saved to add the following lines after the case statement:

```
var=’wlsnams -k @NameDatabase:UserNameDB $username | tr ’,’ ’\n’ |
sed ’s/^.*://g’ | sort -u | wc -l‘
#
if [ $var gt 0 ]; then
echo FALSE
else
echo TRUE
fi
```

The final step is to replace the validation script with the new one using the following command:

```
wputpolm -v @UserProfile:UA-A-L login_name < /tmp/val_login_name
```

In a multi-TMR environment, the possibility remains that the same login name can be specified in separate TMRs. This is avoided if a centralized configuration is used, with a top-level TMR being where all the users are defined and subsequently distributed to lower-level TMRs. (See 3.2.1, “Centralized Administration” on page 32.)

4.2 Implementation Considerations

This section describes some additional factors you need to be aware of during a Tivoli User Administration Installation.

4.2.1 Keep to One Version of Tivoli User Administration

All nodes installed with Tivoli User Administration within a TMR should be at the same level to be considered valid for support purposes. When installing any upgrades or patches, they should be applied to all nodes in the TMR that have Tivoli User Administration installed.

This can cause problems in a large installation where there is little chance of upgrading every one of 200 managed nodes before further user administration can take place. A standard populate or distribution should work across slight variations in release levels. However, if problems occur it will be necessary to get systems to the same level before Tivoli support will be able to help. Check the product release notes for further information

Across TMR boundaries, standard distributions and populates should work but the aim again should be to keep all systems at the same level wherever possible. Actions, such as creating users across TMR boundaries where different versions of Tivoli User Administration, are installed are likely to fail.

4.2.2 Managing Novell NetWare

NetWare users are handled through the GUI like those of any other platform. One particular implementation consideration is that you must run `wsetnds` before you can populate or distribute a user profile from/to NetWare NDS. This command allows you to specify to Tivoli the NetWare account to use to access the NetWare NDS tree.

The `wsetnds` must be run any time the NetWare account changes (such as after a password change).

4.2.3 Populate Considerations

Tivoli User Administration must be installed on the node form which the profile is being populated.

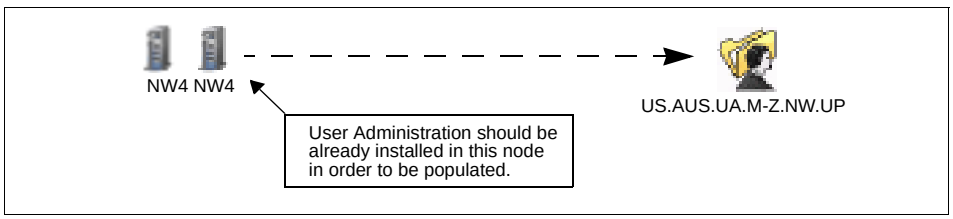


Figure 22. Tivoli User Administration Installed on Source

The populate will fail validation for UNIX UIDs for root and nobody. When you are populating UNIX machines, you can expect some errors from the Populate Errors dialog that warn you that users root and nobody have failed validation policy and have not been added to the profile.

Figure 23. Populate Errors Dialog

In the first entries, the dialog warns you that users root and nobody (as well as the other two users) have failed the validation policy and have not been added to the profile. In fact, the root UID is 0 and nobody UID is -2; the validation policy checks for users with UID greater than 0.

If you want UNIX root or nobody to be included in the populate, you have to disable the validation policies or edit the validation policy script. For more information on how to edit and change the validation policies, see “Altering Validation Policies” on page 69, or refer to the *Tivoli User Administration User and Group Management Guide*.

Windows NT does not export an API call to enable the retrieval of passwords, even in an encrypted form. When you populate from Windows NT nodes and NetWare servers, you won’t be able to get user password information, and therefore, users populated from these endpoints will have their passwords set in the Tivoli database to the login name of the account. When these accounts are distributed to an endpoint where they already exist, their passwords are not changed at the endpoints, unless the password is subsequently changed

in the profile following the populate. If the account does not exist at an endpoint, a distribution of freshly populated accounts will create new accounts with the account's login names as their passwords. Once you have populated from Windows NT or NetWare accounts, the Tivoli administrator can use the GUI or `wsetusr` or `wpasswd` to assign passwords to them.

A further implication that should be considered when populating from Windows NT and NetWare is that a newly populated user cannot change their own password using Tivoli tools, such as `wpasswd` or the OnePassword web GUI (see 5.2.4, "Tivoli Web Password Utility" on page 92), because their old password is not stored in their Tivoli user record. The populated accounts would need to have their passwords explicitly modified by an administrator and re-distributed, so that the endpoint password matched the password in the user record. Changing passwords from endpoint-specific tools is, of course, not affected.

For more information about `wsetusr` and `wpasswd` commands, refer to the *Tivoli User Administration User and Group Management Guide*.

If the system from which you are populating user information contains a very large number of users, it may be necessary to split up the list of users and provide subsets for population. Refer to the Tivoli User Administration documentation regarding this procedure. Different endpoint implementations provide different methods for performing this splitting exercise. This is most likely to be an issue on endpoints such as the OS/390 Security server and AS/400.

4.2.4 Distribute Considerations

Distribute behaviour has changed with different releases (refer to release notes) but distribute with exact copy and all levels can erase system accounts such as nobody, adm, bin, lp, and any non-root UID 0 account, unless they are explicitly defined in a user profile. Tivoli User Administration will not remove root (UID 0) accounts from UNIX or the Administrator accounts from Windows NT or Novell NetWare. Note, however, that the Windows NT and Novell NetWare Administrator accounts are keyed off the English language text name. If you are using some other name for the administrator, that account will be deleted in an all level, exact copy distribution unless the account appears in a user profile. Note that distributions to Tivoli Management Agent endpoints will not result in any such implicit account deletion.

When creating UNIX user's home directories, the TMR server requires root write access to the home directory file system. If a UNIX user's home

directory is to reside on NFS, and the NFS server is *not* a Tivoli managed node, the TMR server will need root write access to the home directory file system in order to set up the home directory.

Tivoli User Administration adds a custom-function that allows you to specify record-level subscription allowing greater flexibility in determining specific records in a profile to be distributed to specific endpoints. The record-level subscription is a Tivoli User Administration implemented function, however AEF action scripts are dealt with through framework interfaces. This means that there may be circumstances where an AEF action will execute because the profile went to a machine, but the action-related record was not distributed to that machine as record-level subscription configured it otherwise. This is not likely to cause any major problems but you may experience messages indicating AEF actions failed.

4.2.5 Cleaning Out file_versions Directory

A periodic maintenance task should be to check, and if necessary, clean up the file_versions directory on UNIX managed nodes.

Tivoli provides file input/output routines for tracking and controlling revisions to system files made by Tivoli applications, including Tivoli User Administration. Each time, for example, that Tivoli User Administration updates the UNIX /etc/passwd file on a system, the previous version is maintained under a revision control system (RCS). You can use the RCS tools provided with Tivoli to recover previous revisions of a system file should problems occur. The versioning of system files by Tivoli User Administration is completely automatic. The updated system file is placed in the standard location on the system with the appropriate structure and contents. The versioned copy of the system file is maintained under the Tivoli database directory in a special directory called file_versions. For example, if your database is stored in the directory `$DBDIR/fuji.db`, the versioned file for the /etc/passwd system file is located in:

```
$DBDIR/fuji.db/file_versions/etc/RCS/passwd,v
```

You can determine the database directory by using the `odadmin` command or by looking at the `$DBDIR` variable once `setup_env.sh` has been run. Occasionally, you should prune the RCS log for the versioned system files generated by Tivoli User Administration and other Tivoli applications. After a period of several months, the versioned password file might contain several hundred deltas reflecting the changes that have been made over the elapsed period of time.

Refer to the discussion on versioned system files in the *Tivoli Framework Planning and Installation Guide* for more information regarding clearing up the file_versions directory with RCS commands, such as `wrlog` and `wrcs`.

4.3 Broadening Supported Platforms

Tivoli User Administration is fulfilling the goal of a central interface to manage a wide variety of IT user data. The base product is continually expanding to include new, fully integrated endpoints, such as LDAP, OS/400, and the OS/390 Security Server. In addition, there are many other add-on products being made available to manage user data for a variety of applications including Lotus Domino/Notes, various database systems, firewall products, and single sign-on (SSO) solutions.

This section discusses a few of these areas as an overview of the ways in which user administration can extend to cover most IT aspects. This is not intended to be an exhaustive coverage of the topic, but instead, an introduction to the culture of managing anything to do with users, anywhere, with a single management interface.

4.3.1 Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) is a directory service protocol that runs over TCP/IP. It is designed to provide access to an X.500 directory, while not incurring the resource requirements of the complete Directory Access Protocol (DAP).

Many vendors of products that include user and directory information are adding LDAP connectivity. The types of systems manageable through LDAP include:

- Email systems, so user mail accounts can be managed along with system accounts.
- Native LDAP directories, such as Netscape Directory Server.
- Existing directory servers that have added LDAP support to their native protocols, such as X.500 directories and Novell's NDS.

LDAP is based on the client-server model. One or more LDAP servers contain the data that makes up the LDAP directory tree. An LDAP client connects to an LDAP server and sends a request. No matter to which LDAP server a client connects, it accesses the same view of the directory (depending upon the user's authorization); thus, one LDAP server references the same entry in the directory tree as another LDAP server.

In Tivoli User Administration 3.6 Tivoli have introduced the capability to manage LDAP. In the 3.6 release, LDAP support is being provided as a “technology preview”. This means the code is provided on an “as is” basis allowing you to experiment with it and use it at your own risk. The intention is to provide fuller support in the future.

To manage LDAP user accounts with Tivoli User Administration, you must first establish an LDAP connection (a Tivoli managed resource) and associate it with an LDAP accessible service. The LDAP connection specifies properties, such as the IP address of the LDAP server host and port number of the LDAP accessible service.

The connection also specifies a class name for objects to be managed. As LDAP models can vary, a class is used to define a collection of attributes appropriate for that model. These attributes form the basis of a mapping between Tivoli User Administration user profile attributes and the LDAP attributes.

4.3.2 Database Integration

Implementations of the database management modules, current at the time of writing, used their own user profile implementations to manage database users. Future Tivoli products may well use Tivoli User Administration to perform database user management.

4.3.3 OS/390 Security Server Support

In releases of Tivoli User Administration up to 3.6, support of users of OS/390 mainframe systems was provided through components of Tivoli Global Enterprise Manager (GEM) and the OS/390 Security Server.

The Tivoli Global Enterprise Manager (GEM) product is intended as a bridge between the OS/390 host environment and the distributed environment, consisting of UNIX, Windows NT, Novell NetWare, and so on, to facilitate systems and network management.

The OS/390 Connection Service is the MVS component that responds to requests originating from the Tivoli object dispatcher (the oserv). It is implemented as an MVS address space and contains the TCP/IP protocol support to manage the communications link between the Transaction Program (TP) server and Tivoli.

The TP server, a component of the Tivoli Global Enterprise Manager User Administration Service, responds to requests from the OS/390 Connection Service. It also manages the communication link between the TP server on

the OS/390 server and the OS/390 Connection Service using TCP/IP as a transport mechanism. Data flowing between the Tivoli workstation and OS/390 environments is encrypted, providing a secure environment for transmitting and managing user administration requests. TP server's processing includes obtaining a session key for encrypting and decrypting the data and mapping the Tivoli administrator user IDs to OS/390 user IDs.

The OS/390 Connection Service administers security tasks across multiple platforms with a single action. It manages SAF-compliant products, such as RACF and Computer Associates' ACF/2. By using OS/390 Connection Service, you will have increased security through faster updates and consistent administration of policy. OS/390 Connection Service eliminates the need for administrators to log on to different systems. OS/390 Connection Service provides link encryption and authenticates requests from Tivoli to ensure that they originate from authorized users.

Based on the request, the OS/390 Connection Service will invoke MVS system services, such as SAF, to process the request.

Any number of TP servers can be running on the OS/390 server, but each TP server requires a unique TCP/IP port. When making a connection, the TP server is uniquely identified by specifying the server name and the port number. Each TP server supports multiple users, with each request being processed asynchronously.

The new method for managing OS/390 integrates the features of the Tivoli Global Enterprise Manager. This means GEM is not a required component and instead the OS/390 is treated in the same way as any other endpoint. The Tivoli Management Agent for OS/390 provides the mechanism for distributing user profiles to OS/390. Tivoli User Administration also has an extension available to include user administration support for the OS/390 Tivoli Management Agent.

The new component structure is as follows:

- Tivoli Management Agent on OS/390:
  - Code in SMP/E format on 3480 tape cartridge as its own product.
  - Code integrated into OS/390 V2R7 base.
  - Patch to Tivoli Management Framework (TMF) for OS/390 binaries for the Gateways.
  - Program Directory
- Application Development Environment (ADE) updates in support of the TMF enhancements:

    - OS/390 ADE code to be placed on a re-cut TMF/ADE CD.
- Event Integration Facilities (EIF) services (Tivoli Management Architecture enabled) for OS/390:
    - OS/390 code in SMP/E format on 3480 tape cartridge to be included in the TEC package. The existing OS/390 EIF Cartridge will be rebuilt to contain both the secure and non-secure versions of EIF.
    - Program Directory.
- User Administration Agent on OS/390:
    - OS/390 code in SMP/E format on 3480 tape cartridges as its own product.
    - Server-side code packaged as a separate CD containing AEF-based enhancements for Tivoli User Administration.
    - OS/390 Supplement to Tivoli User Administration document libraries.
    - Program Directory for the OS/390 install.
    - Release Notes for the server side code.
- Security Management Agent on OS/390:
    - OS/390 code in SMP/E format on 3480 tape cartridges as its own product.
    - Patch to Tivoli Security Management for the server-side binaries for OS/390 Security Server support.
    - Server-side code packaged as a separate CD containing AEF-based enhancements for Tivoli Security Management.
    - Program Directory for the OS/390 install.
    - Release Notes for the server side code.
    - OS/390 Supplement to Tivoli Security Management document libraries.
    - OS/390 Security Server code for Role support.

4.3.4 OS/400 Support

The IBM AS/400 is a mid-range system running a proprietary, but reasonably open operating system, known as OS/400. As with any system, there is a need to manage user data, and the Tivoli for AS/400 product enables OS/400 as a Tivoli Management Agent with extensions for Tivoli User Administration, Security Management, Inventory, Software Distribution, Enterprise Console, and Distributed Monitoring.

Like the OS/390 endpoint, OS/400 adds a large number of attributes to user profiles - 72 attributes arranged in 10 new subcategories. The endpoint supports population of user profiles, as well as distribution and passwords that can be managed with the Tivoli User Administration password tools, such as `wpasswd`.

The AS/400 endpoint has a few extra features, such as its own delete action options, that are more compatible with OS/400 `DLTUSRPRF` actions, as well as the ability to control AS/400 group membership attributes. A script is provided, called `w4getusers.pl`, to help manage user profiles in smaller blocks (see also 3.4.3, “User Administration Profiles” on page 56). This script produces files containing lists of user names that can be used as input to the `wpopusrs -f` command.

4.3.5 IBM Global Sign-On

Tivoli GSO User Administration is an extension of Tivoli User Administration, providing the capability to manage IBM Global Sign-On (GSO) user accounts. IBM GSO allows authorized users to sign on only once to gain access to multiple servers, hosts, applications, and systems found in today’s diverse, distributed environments. The extension is provided on the IBM GSO Server CD from release 1.5 onward.

4.3.5.1 GSO Prerequisites

The Tivoli GSO User Administration extension, at the level current at the time of publication (1.5), can be installed on Windows NT or on AIX or Solaris UNIX systems. GSO, itself, also requires DCE. Refer to the product documentation for detailed installation requirements.

4.3.5.2 GSO User Administration Structure

End users continue to use GSO to view their GSO logins, passwords, and targets. They also change their passwords from GSO. From the end-user perspective, the one difference when using this extension is that they cannot administer their own targets. That function is now part of the central management.

The GSO extension works by extending the Tivoli User Administration user profile. This allows manipulation of GSO attributes, through the GUI, or using the standard Tivoli User Administration and Tivoli Management Framework commands. The profile can then be distributed to a new type of subscriber, a GSO cell.

The delegation of GSO administration through Tivoli is governed in the same way as other endpoints, through the use of the framework admin and senior roles and policy regions. In addition, user profiles can be populated with information from GSO cells, although in the current release, this can only be achieved through the command line.

4.3.5.3 GSO Tivoli Components

The Tivoli GSO User Administration extension must be installed on a system that is already installed as a managed node, and a GSO Server after Tivoli User Administration has also been installed. There are two components for installation via the framework:

- Tivoli GSO Cell Service
- Tivoli GSO User Administration Service

The Tivoli GSO Cell Service installs on the TMR server and adds the GSO Cell as a managed resource. The Tivoli GSO User Administration Service installs on Tivoli clients (managed nodes) and enables the communication between the managed node and the GSO Server.

Once installed, you must create GSO Cell managed resources for each GSO Server, which can then become subscribers to Tivoli User Administration profiles. You will use the same considerations for placement within policy regions as for other resources (see 3.2, “Logical Design” on page 32). Remember that unless you separate GSO and platform user profiles (such as Windows NT), a GSO administrator will be able to modify the platform attributes of GSO users. There will be a new user administration category, called GSO, and two new properties with which to configure GSO attributes: GSO Login and GSO Targets. Note that version 1.5 does not include any default or validation policies, although the product documentation includes information about the target data should you wish to write your own.

The GSO Login attributes are keyed into the remainder of the user records through the common login field. The common password can also be used for GSO accounts, or a separate GSO password can be specified. The other attributes are as follows:

GSO User ID The DCE user ID the GSO user will use to gain access to their services, systems, and applications.

Description Free-form description field.

Account Status Active or inactive.

Create DCE Account? Check box used when a DCE account does not already exist for this user to instruct GSO to create a DCE account. (Note that if you are using the santix DCE management software in Tivoli User Administration, that should be used to create DCE accounts and not GSO.)

DCE Password	Password to use (common password used if this is not specified). Password is also verified in a second field.
Authorize	Button used to authorize the creation. Must be done by an administrator with the senior role.

The GSO Targets information is again keyed through the common login. The target attributes are as follows:

Target Type To Add	Selecting a supported target type from the list will result in the display of the Create Target dialog.
Target Type To Edit	Selecting an existing target will result in the display of the Edit Target dialog.
Remove a Target	Presents a dialog of existing targets from which one can be selected for deletion.

4.3.5.4 Linking Passwords

It is likely that if you already have Tivoli User Administration in use before adding the GSO extension, you will be managing user accounts on systems also covered by GSO. The GSO extension allows you to link those passwords to the GSO configuration. For example, if you have a user profile for a user on a Windows NT Server, you can link the password attribute in that user's Windows NT profile into the GSO record for the same user. Then, if you modify the password in the Windows NT profile, the GSO record will also be updated. The link can be to the current or separate profile.

> **Note**
>
> When using GSO, the order in which you distribute a profile to its subscribers is important. GSO should always be the last subscriber to which you distribute profiles.

4.3.5.5 GSO Cell Management

From a GSO Cell resource icon on the Tivoli desktop, you can log on to the cell server. This logon starts a daemon that acts as the communications link between GSO and Tivoli and authenticates the daemon to DCE. You can then check the cell status (whether it is up or down), configure passticket server information (typically used by host access control products such as RACF), or update the target type list.

4.3.6 PassGo Technologies' PassGo Single Sign-On

PassGo (formerly CKS) also has an SSO solution that integrates with Tivoli User Administration. The product is called PassGo MyNet SSO. The MyNet SSO product provides a single sign-on capability through a client application desk providing logon programs for OS/2, DOS, and Windows 3.x, 95, and NT with agents for the platforms supporting those clients, including the major network operating systems.

The MyNet model provides user authentication, including the use of additional verification systems, such as smart-cards. Of course, all user data transferred by MyNet is encrypted, and auditing provides data on a user, system, or time basis.

PassGo Technologies has extended Tivoli User Administration to support administration of the SSO User Profiles on the MVS Authentication Server. This means that administration of User information for SSO is now administered in the same central graphical environment as the respective Access Control system, for example, RACF and UNIX.

It is, therefore, not necessary to logon to PassGo's MVS Authentication Server via the 3270 interface to administer SSO User Profiles.

The extension of Tivoli User Administration is achieved using AEF and ADE, delivering a highly integrated combination of PassGo and Tivoli User Administration.

Security is provided between all components of the system. For example, data is encrypted across the IP link to MVS and APF authorization for the MVS components will prevent spoofed Admin commands.

PassGo Technologies has written a new MVS Tivoli Intelligent Agent that is responsible for processing Tivoli User Administration administration in conjunction with the PassGo Administration API. This Agent is known as the Tivoli Information Router (TIR).

The TIR is designed to be extensible and support administration of non-PassGo related information. With the customizing of the Tivoli User Administration graphical front end, new administration entities will be passed across the Tivoli/PassGo network components to the TIR. The TIR, in turn, will drive an alternative API or Access Method allowing a third party MVS database to be updated from the central graphical Tivoli User Administration environment.

4.3.7 Check Point FireWall-1

An effective firewall must track and control the flow of communication passing through it; simply examining packets is not sufficient. FireWall-1 uses an architecture Check Point called Stateful Inspection Technology. This allows FireWall-1 to make control decisions using the following:

Communication Information	Information from all seven layers of the OSI networking model that is contained in the packet.
Communication-Derived State	The state derived from previous communications, such as saving the outgoing PORT command of an FTP session, so that an incoming FTP data connection can be verified against it.
Application-Derived State	The state information derived from other applications.
Information Manipulation	The evaluation of flexible expressions based on all the above factors.

FireWall-1 provides encryption, logging, and alerting mechanisms, and most relevant to Tivoli User Administration, authentication. This is defined through the use of a FireWall-1 Security Policy - a way of implementing network protection measures from an organization's full security policy (for a discussion on security policy, see the *Tivoli Security Management Design Guide* Redbook).

FireWall-1 defines firewalls, services, users, resources, and the rules that govern the interactions between them. This is implemented through firewall modules containing Inspection Code installed on gateways, hosts, routers, switches, or packet filters, enforcing the FireWall-1 Security Policy.

Firewall-1 already has a number of integration points with Tivoli, through monitoring collections, for example. From a user administration perspective, this will be expanded through an implementation of LDAP on FireWall-1 allowing user data to be managed from Tivoli User Administration. (See "Lightweight Directory Access Protocol" on page 76.) Other vendors are integrating their firewall products with Tivoli, including IBM.

Note

There are special considerations when implementing Tivoli across firewalls or adding firewalls to a Tivoli implementation. The Fall 1997 edition of *The Managed View* (published by Wellesley Information Services, Inc., Newton, MA, 02159, USA) has a good article on the subject. Consult your Tivoli support contact for more information and advice.

4.3.8 Lotus Domino and Notes

The optional Admin Extension for the Tivoli Manager for Domino product adds Notes-specific components to the user profile managed resource of Tivoli User Administration. These added components enable an administrator to add Notes Configuration, Login, Mail, and other information for Notes users into a Tivoli user profile. This profile can then be distributed to any subscriber specified in the Profile Manager. Only Notes server endpoints are valid endpoints for Notes user information.

The administrator can also add non-Notes information (such as platform information) for these same users to the same user profile. When the profile is distributed to subscribers, actions take place depending on the type of endpoint the subscriber represents. For example, a user profile dropped on a Notes server endpoint pushes only the Notes server information (such as Name & Address book updates) to that endpoint. The same profile dropped on a UNIX managed node pushes only UNIX information (such as user accounts) to that endpoint; any Notes databases that may reside on the UNIX managed node remain unchanged by the profile, since only Notes server endpoints are valid endpoints for Notes user information.

Configuring the admin extension creates a new Notes category in user profiles with four subcategories: Configuration, Login, Mail, and Other Information.

Chapter 5. Operations and Maintenance

Once Tivoli User Administration is installed and running, there are a number of operational activities likely to be required. This chapter documents those activities and process information likely to be of use in the live environment.

5.1 Skills Transfer to New Administrators

It's important to transfer skills to the administrators that will finally work with the day-to-day operations. As they will be the ones dealing with problems, updates, and administration, you need to be sure they will be adequately prepared.

Once setup has been completed, a person with even a limited understanding of the operating system-specific account maintenance process will be able to assume responsibility for on-going account management. This would include creation, deletion, password resets, and so on.

For instruction in Tivoli User Administration, prerequisites may include a working knowledge of native user administration for Windows NT, UNIX, NIS, some experience in shell programming, and an understanding of the Tivoli Management Framework, specifically:

- Use of the Tivoli desktop
- Populating an administrator's desktop
- Use of policy regions
- Creation of profiles and profile managers
- Distribution of profiles to subscribers
- Creating and using tasks, task libraries, and scheduled jobs
- Understanding the concept of authorization roles within Tivoli

Tivoli offers courses that provide these skills, as well as courses explaining the Tivoli User Administration product.

If you found during the gathering information phase of the project that the customer doesn't have a very well defined user management process, you can use the skills transfer to implement or make changes to a process.

5.2 Managing Passwords

There are several ways to change user passwords in a Tivoli User Administration environment. Both system administrators and users can use Tivoli User Administration tools to manage passwords.

As an administrator, you can use the following tools:

- Using Tivoli GUI
- Using the `wpasswd` command
- Using the `wsetusr` command
- Using the new web password utility (OnePassword)

As a user, you can use the following tools:

- Using local password utilities (platform-provided)
- Using `wpasswd` command
- Using the new web password utility (OnePassword)

The following sections discuss these methods.

5.2.1 Tivoli GUI Password Control

Using the Tivoli administration GUI, you can change user passwords in all the platforms for which the user has a login or change the password for individual operating system types.

When you access the GUI to make changes to user properties, you have the option to set a common login and common password that will be used in all the platforms for which the user has an account. If you decide to set a password for a specific operating system, specific passwords will override the common password when both are provided.

It's important to mention that once you have made the password change in the GUI, it doesn't mean that you have updated the user information. You need to distribute the user profile in order to change the attributes in user system files. Even then, the change must reach the final user account system (all levels) either through the actions of administrators at each subscription level (in a distributed administration model) or through an all levels distribution from the central profile.

This type of change is initially changing the profile data wherever that profile resides. Figure 24 shows the Tivoli GUI for the user profile record password attributes.

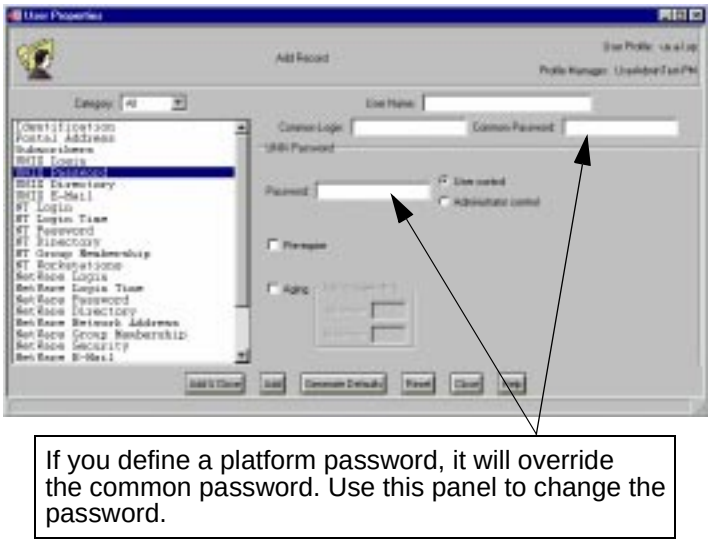


Figure 24. Tivoli User Administration GUI Password Change

5.2.2 Tivoli Command Line Password Control

The `wpasswd` command is used to change a user password from the command line. It is a Tivoli command, so it must be run on a system where Tivoli has been installed.

As with Tivoli GUI, you need to distribute the changes in order to update changes in system files (but see the `-l` option description). A Tivoli administrator with the correct role can use this command to modify any user's password in the Tivoli database, but an ordinary user can only use it to change their own password.

Note

In the 3.6 release of `wpasswd`, a new flag (-L) has been added to perform an 'all-endpoint' change of the password without the need for a distribution of the profile. See the description of the web password utility (OnePassword) for a description of this feature. ("Tivoli Web Password Utility" on page 92)

A word of warning- this is a powerful command that will allow you to change the password for all instances of the user name in the TMR database where the Tivoli Administrator you are using has authorization. In other words, if you have a user named evargas, and you use this command to change their

password, the command will change the password in any profile it finds evargas as a user name. Refer to the Tivoli User Administration User and Group Management Guide for a full description of this command. The following is the Administrator version of the syntax:

```
wpasswd [-e] [-l] [-v] [-tu] [tt] [-tw] [-tc] [username]
```

`-e`	Pre-expires user's password; the user is required to change their password the next time they log in.
`-l`	This command changes the user's password in /etc/passwd file on a local Unix host.
`-L`	Change the password at every this user record is subscribed too (3.6 and beyond only).
`-v`	Provides verbose output.
`-tu`	Changes the password for a UNIX account.
`-tt`	Changes the password for a Windows NT account.
`-tw`	Changes the password for a NetWare account.
`-tc`	Changes the user's common password.
`username`	Specifies the login name of the user.

One example of its use is:

```
wpasswd -l -tu evargas
```

The example above changes the password for the user evargas in a Unix host. The change is also recorded in the /etc/passwd file. And as you can see, since you cannot specify any user profile, the change will apply to any user named evargas in any user profile.

Users can also use the `wpasswd` command to change their passwords. The only requirement is that the administrator should have specified that the user has control of the password. When the user uses this command, the command changes the user's password in all the profiles where this user is defined and where the given current password matches the profile password. And as in the examples above, any change made with this command needs to be distributed in order to take effect.

The user can only use the following syntax for this command:

```
wpasswd [-l] [-L] [-v] [-tu] [-tt] [-tw] [-tc]
```

As you can see from the syntax above, the command assumes the password to change is from the user who invokes the command. The `-l` option performs

an immediate change in the machine from which command was invoked. If the `-l` option is not used, the change will not take effect until the user profile is distributed (unless `-L` is used at 3.6 and above).

One final consideration should be made about the use of this command. It should not be used as a replacement of local system utilities because it can behave incorrectly for any of the following reasons:

- The Tivoli object dispatcher (oserv) is not running, or if you don't have access to the oserv.
- The TMR server not running or accessible.
- The user interrupts the command.
- Different passwords exist locally and in Tivoli.
- Command used by Tivoli administrators without the admin role.
- The system is in single-user mode.
- Command used by Tivoli administrators who do not have admin role for the user profiles.

In some installations, some administrators may try to put a wrapper around the `wpasswd` that takes account of these possible problems and reverts to `passwd`, for example, if `wpasswd` fails.

Note

In the releases current at the time of writing (up to 3.6), there is no integration between `wpasswd` and the Tivoli Security Management `sepass` utility. Password controls in UNIX implemented through `sepass` are not implemented when using `wpasswd`. One option is to add `sepass` into the wrapper described above. Tivoli is looking at integrating these features in a future release.

5.2.3 Tivoli Command Line User Attribute Control

As the user's password is an attribute in the user profile, the `wsetusr` command also can be used to modify the password attribute. This command is the direct command alternative to the Tivoli GUI to change user attributes. It can be used to change most user attributes, as well as common passwords and specific operating system passwords. And as in the GUI, you can choose the user profile where the user is located. Since this command can be used to change other attributes besides the password, we will only look at the options that apply to password changes.

The four variations we will show apply to changing passwords. The first one applies to changing common passwords, the second to changing UNIX passwords, the third to changing Windows NT passwords, and the fourth to changing NetWare passwords.

To change common password use:

```
wsetusr -cp password @UserProfile:profilename username
```

To change UNIX password use:

```
wsetusr -p password @UserProfile:profilename username
```

To change NT password use:

```
wsetusr -pt password @UserProfile:profilename username
```

To change NetWare password use:

```
wsetusr -pw password @UserProfile:profilename username
```

As in the GUI, the change will remain inactive until you distribute the user profile to system files. You can use the command `wdistrib` with the `-l maintain` option to maintain local modifications and with the `-m` option to distribute to all levels of subscribers.

5.2.4 Tivoli Web Password Utility

Still under development at the time of writing, the web password facility (tentatively named OnePassword) provides users and administrators with a web interface to user password administration.

In Tivoli User Administration 3.6 Tivoli will introduce the OnePassword tool as a “technology preview”. This means the code is provided on an “as is” basis allowing you to experiment with it and use it at your own risk. The intention is to provide fuller support in the future.

The effect of using this facility is similar to using `wpasswd` command. OnePassword will change the user profile record, it then initiates the building of a table of endpoints where the profile has been distributed, and finally, it contacts each of those endpoints to make the password change immediately. This is the same behaviour as the new (in 3.6) `-L` option for `wpasswd`.

Connectivity to OnePassword is through the oserv web server on port 94 of the TMR server (http://tmripaddress:94). The first screen from the oserv will show you the available Tivoli Applications and will ask you to choose one of them. When choosing Tivoli User Administration, you have the opportunity to log in using a Tivoli administrator ID and password. If you log in as a Tivoli

administrator with sufficient role definitions, you will be able to specify any common login name to select which user's password to change. If you are not a Tivoli administrator, you will be only able to change your own password.

Figure 25 shows the OnePassword interface (note, this was subject to change at press time).

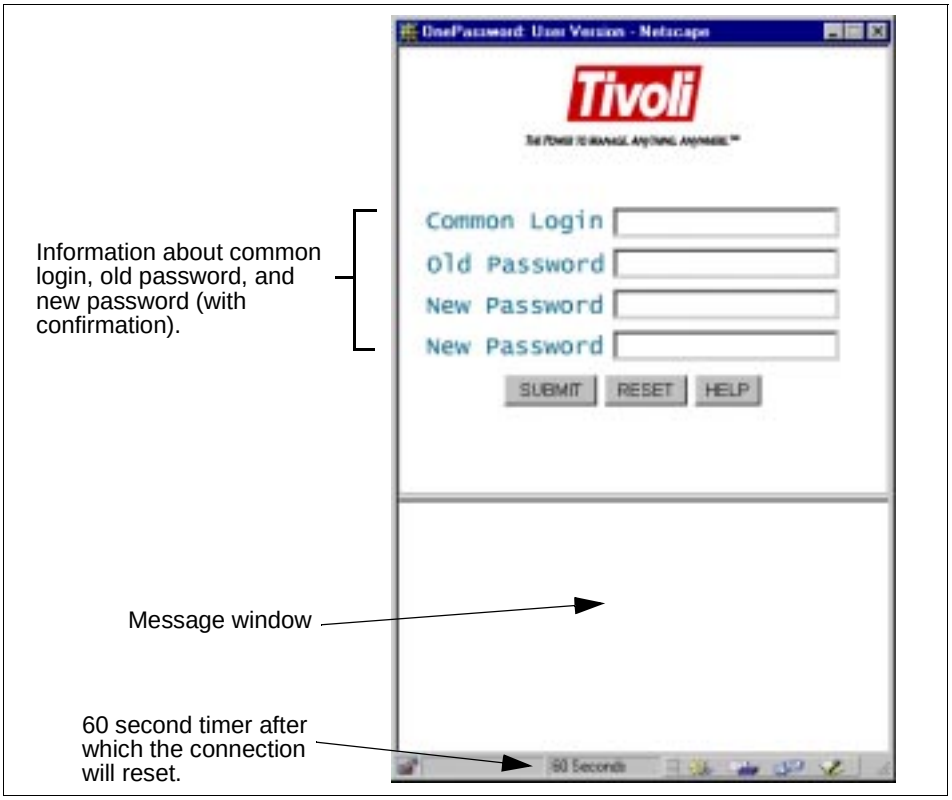


Figure 25. OnePassword Web-Based Password Tool

As you can see from Figure 25, this facility only allows you to change passwords based on the common login. So, you cannot use this to modify the password based on a specific platform, as you can with `wpasswd`.

Communication with OnePassword is through Secure Socket Layer (SSL) to ensure encryption of the password data.

5.2.5 System Specific Password Utilities

As a user, you can use the specific utilities each operating system has as long as they have not been disabled through the Tivoli GUI. For example, a Tivoli user administrator can deselect the **Allow user to change password** in the NT Password properties, and this will be reflected once the profile is distributed to the Windows NT server.

If the user is in control of the password, distribution of user profiles will not overwrite their password changes at the profile level, at the subscriber level, or at the system files level, unless the administrator modifies the password. An explicit change to the password in the user record will update the password at the endpoint.

5.3 Product Maintenance

The user administrators will need to be aware of product updates. There may be a Tivoli administrator who has the dedicated task of adding patches, but if this is the user administrator, they will need the necessary roles to install product patches. See also 4.2.1, “Keep to One Version of Tivoli User Administration” on page 72, regarding the need to maintain the same level of the product on all systems in the TMR.

One other area administrators need to be aware of with Tivoli User Administration Updates is the possibility of local custom changes being overwritten by a patch install or product upgrade. The following rules and notes will help avoid problems relating to product maintenance:

1. Never move or delete any Tivoli dialog pieces.

   A patch may attempt to change or add to the Tivoli User Administration dialogs. If Tivoli pieces are removed, it can cause problems in the application of the patch and/or the subsequent operation of Tivoli User Administration. All customizing should be performed by adding new items and modifying them.

2. Distribute all profiles before a product update.

   Certain types of alterations that can occur in patches and upgrades can cause a profile that was modified with the previous level, but not distributed to change in such a way that it will not be distributable after the patch is installed. As distributing a profile with no changes has very little overhead, it is recommended to distribute all profiles before applying updates to ensure that all changed profiles have been distributed.

3. An update will not affect custom AEF attributes - *except*...

Watch the release notes, but a product update should not touch any custom AEF attributes or action scripts. The one exception could be during a major release update, as we had from 2.x to 3.x, and as could happen from 3.x to 4.x. The release notes are where such issues would be documented.

4. Default and Validation policy are never overwritten.

   Modification to Tivoli User Administration policies are documented in release notes but are not applied to existing installations in case local modifications have taken place.

5. Always check the release notes.

   The release notes will highlight any major changes and any likely effects of those changes. Always review them before installing updates.

6. Changing endpoint type after upgrading to 3.6.

   Note that if you change an endpoint (such as UNIX or Windows NT) from a managed node to the Tivoli Management Agent, all scripts used in the management of those endpoints should be checked. Some of the Tivoli `wxxxx` commands will not run on the Tivoli Management Agent, and other tricks that apply to managed nodes may not apply to the Tivoli Management Agent.

5.4 Documenting the User Administration Process

The remainder of this chapter is devoted to user administration process documentation. As someone leading an implementation or responsible for skills transfer, you may become a documentation coordinator whose role is to ensure process documentation meets certain requirements. These could include conformity with process standards, such as ISO9000. Based on IT process models used within IBM, we will propose activities, which provide control and management of user administration process documentation, using a methodology that could be applied to any management process.

It is important that you recognize the potential needs for updates of user administration process documentation, since any change in process can affect organization and technology (such as Tivoli User Administration). Another reason to maintain accurate updates of documentation is that you can ensure people will work in a standard way, with no one setting up their own processes to do the same tasks. Operational areas, such as user administration, are dynamic and need regular revision to avoid becoming outdated. The following activities help keep accurate user administration process documentation and also create new documentation, if required.

The suggested flow includes the following steps:

1. Identify affected process documentation.
2. Assign update/create documentation activity.
3. Perform update/create documentation activity.
4. Verify updated/created documentation.

These steps are broken down in the sections that follow.

5.4.1 Identify Affected Process Documentation

Once you have received a request to update process documentation, you need to establish which parts require updating; therefore, it is important to maintain a register of all the documentation updates. In this register, you will also keep scheduled reviews based (if applicable) on ISO9000 needs.If a request is received for which no current documentation exists, new documentation will need to be created and registered.

5.4.2 Assign Update/Create Documentation Activity

If new documentation is required, a person must be assigned to this activity. The choice of person is important in order to ensure a process that everyone can understand.

The following skills and resources should be considered when assigning someone to update process documentation:

- Subject knowledge and experience required to translate the information into easily understandable language and terminology that can be understood by all skill levels within a user administration operations group.
- Understanding of the format in which the documentation is held.
- Time available to complete the task within the required time frame.
- Not already maintaining too many other documents.

If the document already exists, information about who updates the document should be found in a document control information section.

If the document needs to have ISO9000 Information, or the document owner/maintainer information changed, this should be carried out by a designated documentation coordinator. Otherwise, you should assign the task to the document maintainer(s). The following information should be supplied to the document maintainer(s):

- The title of the existing/new documentation affected

- The details of the change required
- A specified completion date for the work that is compatible with the urgency to maintain the documentation

5.4.3 Perform Update/Create Documentation Activity

During this activity, the document maintainer(s) will carry out the necessary changes to the existing or new document(s). As the documentation coordinator, you will manage progress on the various tasks assigned, and if the task is sufficiently large, will ask for regular status reports or meetings. The documentation coordinator can then address any problems or issues resulting from these checkpoints.

5.4.4 Verify Updated/Created Documentation

The document maintainer(s) will notify you, as the documentation coordinator, as soon as the tasks have been completed. You must ensure the work satisfies the following criteria:

- The tasks specified have been addressed and completed.
- The document control information has been suitably updated.
- The updated document is stored, and any old versions are removed from the database.
- The documentation register is updated to reflect the changes.
- New documentation must be ISO9000 compliant, if applicable.
- Agreement to the change is secured, if appropriate.
- Communication and education is organized, if appropriate.

5.5 Management process

As we discussed in 1.5, “Process Overview” on page 3, if you have a system management process in place before beginning your implementation, you will find it much easier to implement Tivoli User Administration. Obviously, if you hand over a set of well defined processes that are tightly related to the tool you are implementing (Tivoli User Administration), there will be a greater probability of success in the day to day operations if and when you are no longer in the loop.

In order to help you to develop this process and make the integration with Tivoli User Administration, we will suggest in the following sections a process

which could be followed and developed to establish this relationship between Tivoli User Administration and systems management processes.

Taking the example from “Process Overview” on page 3, we will discuss how our suggested model could be applied to user administration processes using Tivoli User Administration as a tool.

5.5.1 Process Relationships

User administration probably won’t be the only process in place, and it is very likely that Tivoli User Administration will not be the only tool implemented.

Therefore, you will find a lot of relationships between processes, with inputs from other processes and outputs to other processes. In Table 16, you will find the most common relationships that you can find between the user administration process and other processes (including other Tivoli applications in use).

Table 16. Process Relationships - Applications Affecting User Administration

Applications	Application Process Invoking the User Administration Process	Input Received from Invoking Process	Output Returned to Invoking Process
Tivoli Inventory	Moves, adds, and changes	Service request and complete description	Status of request results
Tivoli Information/ Management	Customer services	Service request and complete description	Status of request results
Tivoli Enterprise Console/Help Desk Application	Problem management	Service request and complete description	Status of request results
Tivoli Enterprise Console/Help Desk Application	Change management	Service request and complete description	Status of request results

Table 17 shows how the user administration process may affect other management processes.

Table 17. Process Relationships - Applications Affected by User Administration

Applications	Application Process Invoked by the User Administration Process	Input Sent to Invoked Process	Output Received from Invoked Process
Tivoli Enterprise Console/Help Desk Application	Problem management	Description of problem	Problem resolved or not resolved
Tivoli Information/ Management	Customer services	Request of service	Status of request
Tivoli Enterprise Console/Help Desk Application	Problem management	Request of service	Status of request
Tivoli Enterprise Console/Help Desk Application	Change management	Request of service	Status of request
Tivoli Inventory	Asset tracking	Request of service	Status of request
Tivoli Enterprise Console/Help Desk Application	Change management	Description of change	Change approval or disapproval

5.5.2 Roles and Responsibilities

The roles and responsibilities should be established in accordance to customer preference and existing processes and should include Tivoli User Administration roles. The following are some examples of the main roles and responsibilities that can be found in a user administration process.

5.5.2.1 Tivoli Enterprise Administrator

The Tivoli Enterprise Administrator role is to set corporate-wide policy at the top TMR level (tier 1) with the understanding that this policy must also be set at each subsequent tier level (Tivoli policy will not propagate down the TMR tier levels automatically). This group will be restricted to a few members from each organization. The Tivoli Enterprise Administrator responsibilities include:

- Create policy regions.
- Ensure policy propagation.
- Create administrator accounts.

- Make recommendations for process, procedure, and tool improvements.

5.5.2.2 Tivoli Operations Administrator

The Tivoli Operations Administrator has overall responsibility for adding, changing, moving, and deleting requested user and group access to one or more servers on the network. The Tivoli Operations Administrator responsibilities include:

- Reset network and user ID passwords.
- Add, change, or delete user IDs.
- Add, change, or delete groups.
- Data moves, such as moving files remaining in home directories from deleted user IDs.
- Software access.
- Space allocations.
- Name changes.
- Name access restrictions.
- Login script adjustments.

5.5.2.3 Tivoli Help Desk Administrator

The Tivoli Help Desk Administrator role is to assist the end user in using all Tivoli User Administration tools as provided and to perform password resets. Tivoli Help Desk Administrator responsibilities include:

- Reset network and user ID passwords
- Login script adjustments

5.5.3 Tivoli User Administration Process

Now, we are going to show you how information flows in our suggested user administration process. Note also, that even the most sophisticated management tools still require some manual intervention in cases such as verification of the validity of a user request.

Once we have defined the flow of the process, we need to describe it, adding objectives, inputs, and outputs from other processes, roles, tools, and prerequisites. This is an important task, since this information will be shared with the administrators who will need to follow it and maintain any changes in the documentation.

Let’s begin with the flow that describes the full User Administration process shown in Figure 26.

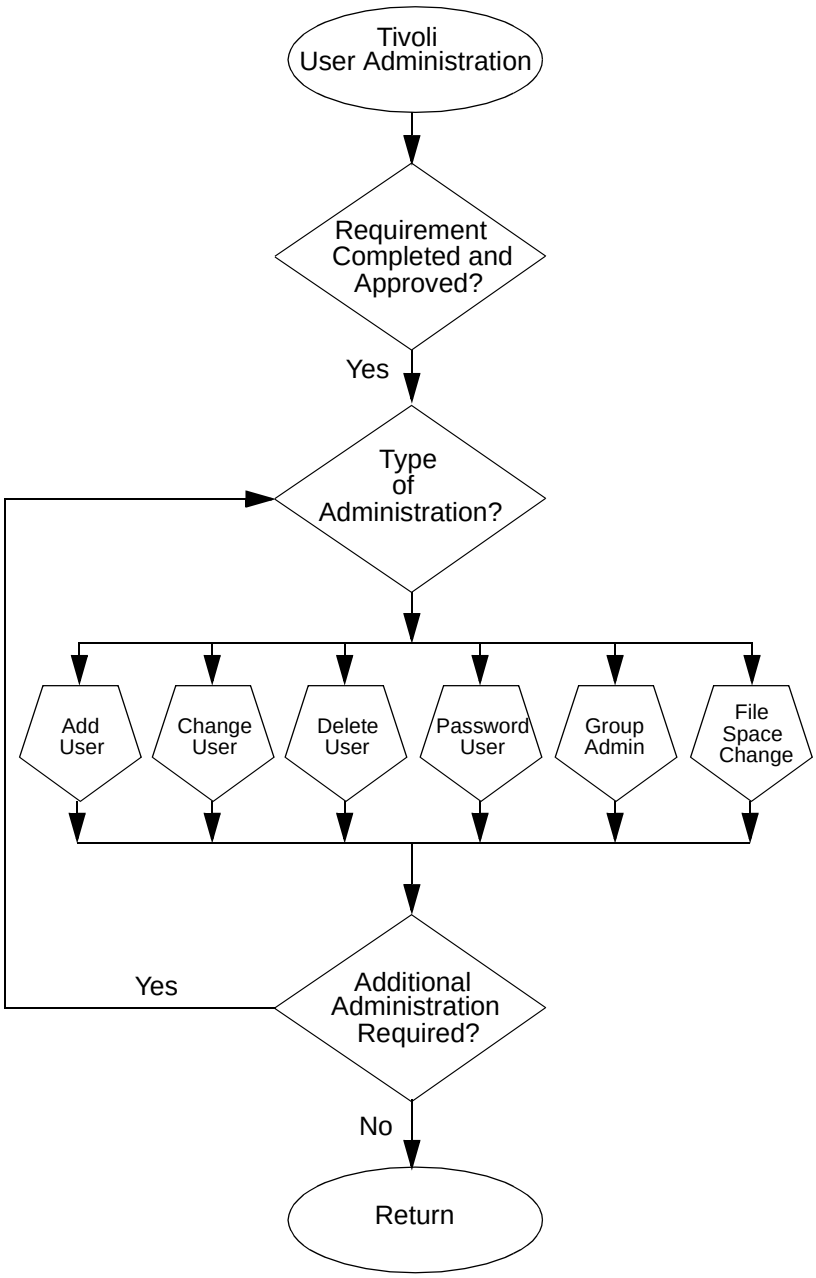


Figure 26. Tivoli User Administration Process Flow

5.5.3.1 Tivoli User Administration Flow Description

The flow can be described in more detail as follows:

Objective	To administer user profiles and profile managers by adding, changing, and deleting access to servers and applications.
Roles	Tivoli Enterprise Administrator Tivoli Operations Administrator Tivoli Help Desk Administrator
Tools	Tivoli User Administration
Prerequisites	None
Inputs	Service request from external operational process
Output	Updated documentation and service request record
Subprocesses Called	Add User / Change User / Delete User / Password Reset / Group Admin / File Space Change
Called by Subprocesses	None

The *Tivoli User Administration Process* suggested includes the following steps:

1. Is the requirement completed and approved?
   - If Yes, proceed to Type of Administration?
   - If No, exit to Handle Requirements not Approved.
2. Handle Requirements not Approved.
3. Type of Administration?

Determine the type of administration required by the service request:

- If Add User, proceed to Add User.
- If Change User, proceed to Change User.
- If Delete User, proceed to Delete User.
- If Password Reset, proceed to Password Reset.
- If Group Administration, proceed to Group Administration.
- If File Space Change, proceed to File Space Change.

4. Add User (Subprocess).
   - Proceed to Additional Administration Required?

5. Change User (Subprocess).
   - Proceed to Additional Administration Required?
6. Delete User (Subprocess).
   - Proceed to Additional Administration Required?
7. Password Reset (Subprocess).
   - Proceed to Additional Administration Required?
8. Group Administration (Subprocess).
   - Proceed to Additional Administration Required?
9. File Space Change (Subprocess).
   - Proceed to Additional Administration Required?
10. Additional Administration Required?
    - If Yes, proceed to Type of Administration?
    - If No, proceed to Return.
11. Return.

Now, as an example, we are going to describe in more detail a couple of the subprocesses (Add User and Delete User) of this process. The first flow we will show in Figure 27 is the Add User subprocess.

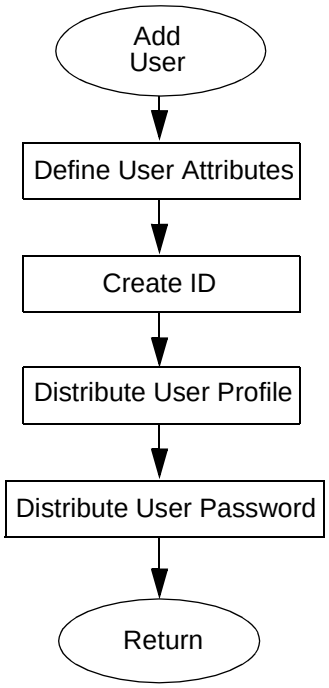


Figure 27. Add User Subprocess Flow

5.5.3.2 Add User Flow Description

The flow can be described in more detail as follows:

Objective	To establish an account for the user in the available systems
Roles	Tivoli Operations Administrator
Tools	Tivoli User Administration
Prerequisites	None
Inputs	Approved request to access shared applications and printers
Output	Successful addition of user ID and password
Subprocesses called	None
Called by processes	Tivoli User Administration

The Add User Process suggested includes the following steps:

1. Define user attributes, such as:

- Login
- Login name
- User ID
- Password
- Home directory
- Group membership
- Network address
- Security
- Mail

and so on, depending on whether you are creating a UNIX, Windows NT, NetWare, OS/390, or other account.

2. Create ID with all the definitions above in a user profile in a suitable profile manager.
3. Distribute User Profile, defining:

Distribute to:

- Next level of subscribers, or
- All levels of subscribers

Distribution type:

- Preserve modifications in subscriber's copies of the profile.
- Make each subscriber's profile an exact copy of this profile.

4. Distribute User ID and Password to User.
5. Return.

Now, we'll take a look at the delete user subprocess flow. This is given in Figure 28.

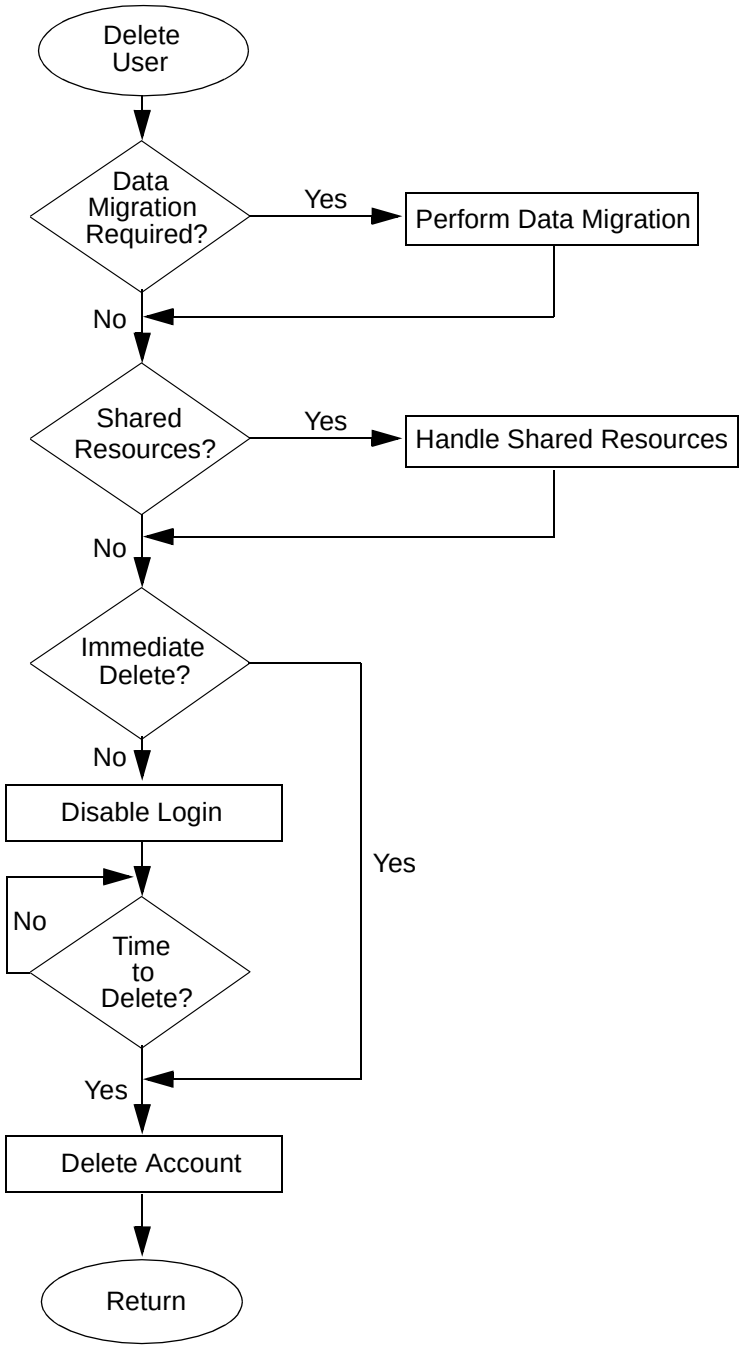


Figure 28. Delete User Subprocess Flow

5.5.3.3 Delete User Flow Description

The flow can be described in more detail as follows:

Objective	To delete user from systems
Roles	Tivoli Operations Administrator
Tools	Tivoli User Administration
Prerequisites	None
Inputs	Approved request for delete
Output	Completed and documented delete request
Subprocesses called	None
Called by processes	Tivoli User Administration

The Delete User Process suggested includes the following steps:

1. Data Migration Required?
   - If Yes, proceed to Data Migration.
   - If No, proceed to Shared Resources?
2. Preform Data Migration.

   This should be done if you need to move the data before deleting the account.
3. Shared Resources?
   - If Yes, proceed to Handle Shared Resources.
   - If No, proceed to Immediate Delete?
4. Handle Shared Resources.

   When the user has shared resources attached (such as scanners, printers, and so on), you may need to use OS-specific tools to detach these shared resources before deleting the account.
5. Immediate Delete?
   - If Yes, proceed to Delete Account.
   - If No, proceed to Disable Login.
6. Disable Login.

   Using the user profile properties window in Tivoli User Administration.
7. Time to Delete?
   - If Yes, proceed to Delete Account.
   - If No, return to Time to Delete?

8. Delete account using the User Profile Properties window in Tivoli User Administration.
9. Return.

By now, you should have a reasonable idea of how this works!

Appendix A. Project Management

This publication cannot hope to tackle the broad and complex subject of project management. However, this appendix contains a little information that will assist those working on the project from a planning point of view.

A.1 Planning User Administration Implementation

A good design will make the implementation much easier. For most environments, the implementation of Tivoli User Administration will be a reasonably large project. While we cannot hope to cover project management topics in any detail in this publication, we will explain some guidelines for using a methodology to breakdown the work required. We also will develop a User Administration implementation plan and milestones for that plan.

We can look at this phase as a distinct project for which we must gather together material including any work performed along the lines of the descriptions in the previous chapters. The outline of the project methodology is as follows:

- Gather all project-related materials.
  - Scope definition
  - Requirements baseline
  - Technical solution design proposals
- Review breakdowns of work used for similar implementations, if available.
- Prepare a summary high-level breakdown of the work required (see “Sample Work Breakdown” on page 112). This should define the overall tasks in pieces that are individually manageable and trackable.
- Develop the work breakdown to identify all the necessary subtasks and individual control items (see “Task Plan” on page 112).
- Involve the responsible project team member in developing the work breakdown. This will be someone, such as a customer project owner, customer IT person, and so on.
- Include project support tasks, such as project management and quality assurance.
- Agree the breakdown of work with customer project contact.
- Add appropriate package to manage risks.

Making a plan can start with something, such as a brain storm session. Then, group similar tasks and work through them until you have a listed breakdown

of the work involved that can be translated into a plan. Even if you are only responsible for implementing a solution someone else designed, you can follow these steps.

A.1.1 Sample Work Breakdown

A brainstorm session and subsequent grouping of tasks may produce the following breakdown of the major steps required in the implementation of Tivoli User Administration:

- Project plan.
- Design and documentation.
- Install Tivoli User Administration on TMR server.
- Define policy regions.
- Define profile managers.
- Define user profiles.
- Install Tivoli User Administration in managed nodes.
- Populate profiles.
- Distribute profiles.
- Determine change procedures.
- Train administrators.
- Translation of activities and responsibilities to administrators.

Now, you need to put these tasks in a plan and establish a description and grouping of sub-tasks for every task you have identified.

A.1.2 Task Plan

An example of one Tivoli User Administration implementation plan could be as follows:

- Project management
  - Project plan, including items, such as resource allocation and milestones.
- Design and documentation
  - Define policy regions.
  - Define profile managers.
  - Define user and group profiles.
- Installation and configuration

- Install Tivoli User Administration on TMR server.
- Install Tivoli User Administration on endpoints.
- Define administrators.
- Create profiles.
- Populate profiles.
- Make initial profile modifications.
- Distribute profiles.

- Hand-over
  - Determine and document change and problem management procedures.
  - Train administrators.
  - Hand over of activities and responsibilities to administrators.

Appendix B. Special Notices

This publication is intended to help those implementing TIvoli User Administration to plan and perform an orderly and successful implementation. The information in this publication is not intended as the specification of any programming interfaces that are provided by Tivoli User Administration. See the PUBLICATIONS section of the IBM/Tivoli Programming Announcement for Tivoli User Administration for more information about what publications are considered to be product documentation.

References in this publication to IBM or Tivoli products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM or Tivoli product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one), and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (vendor) products in this manual has been supplied by the vendor, and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have

been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX
AS/400
BookManager
IBM
MVS
OS/2
OS/390
OS/400
RACF

The following terms are trademarks of other companies:

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix C. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redbook.

C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications, see “How to Get ITSO Redbooks” on page 121.

- *Getting Started With Tivoli User Administration*, SG24-2015
- *Managing Access from Desktop to Datacenter: Introducing Tivoli Security Management*, SG24-2021
- *Implementing TME 10 in High Availability Environments*, SG24-2032
- *TME 10 Internals and Problem Determination*, SG24-2034
- Also look out for a new Redbook on *Managing the OS/390 Security Server with Tivoli* - due for publication late 1998.

C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
Tivoli Redbooks Collection (HTML, PDF)	SBOF-6898	SK2T-8044
Lotus Redbooks Collection	SBOF-6899	SK2T8039
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at `http://www.redbooks.ibm.com/`.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Redbooks Web Site on the World Wide Web**

  `http://w3.itso.ibm.com/`

- **PUBORDER** – to order hardcopies in the United States
- **Tools Disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLCAT REDPRINT
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BokkManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type the following command:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
  ```

- **REDBOOKS Category on INEWS**
- **Online** – send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbpieces become redbooks, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** – send orders to:

	IBMMAIL	**Internet**
In United States	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** – send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** – send orders to:

United States (toll free)	1-800-445-9269
Canada	1-800-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 408 256 5422 (Outside USA)** – ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name | Last name

Company

Address

City | Postal code | Country

Telephone number | Telefax number | VAT number

☐ Invoice to customer number

☐ Credit card number

Credit card expiration date | Card issued to | Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

List of Abbreviations

ADE	Application Development Environment
AEF	Application Extension Facility
CIO	Chief Information Officer
CLI	Command Line Interface
DAP	Directory Access Protocol
DCE	Distributed Computing Environment
DNS	Domain Name Server
DSL	Dialog Specification Language
EIF	Event Integration Facility
GEM	Global Enterprise Manager
GID	Group Identification Number
GSO	(IBM) Global Sign-On
GUI	Graphical User Interface
IBM	International Business Machines Corporation
IT	Information Technology
ITSO	International Technical Support Organization
LCF	Lightweight Client Framework - Now known as Tivoli Management Agent
LDAP	Lightweight Directory Access Protocol
NetBEUI	NetBios Extended User Interface
NetBIOS	Network Basic Input Output System
NFS	Network File System
NIS	Network Information Service
OID	Object Identifier
PCMN	PC Managed Node
RACF	Resource Access Control Facility (major security component of the OS/390 Security Server)
RCS	Revision Control System
SSO	Single Sign-On
TCP/IP	Transmission Control Protocol/Internet Protocol
TIR	Tivoli Information Router
TMA	Tivoli Management Agent
TMR	Tivoli Management Region
UID	User Identification Number
WAN	Wide Area Network

Index

Symbols

A

C

D

E

F

G

H

I

L

M

U

V

W

ITSO Redbook Evaluation

Tivoli User Administration Design Guide
SG24-5108-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
_ **Customer** _ **Business Partner** _ **Solution Developer** _ **IBM employee**
_ **None of the above**

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

_____

_____

_____

_____

SG24-5108-00
Printed in the U.S.A.