HP OpenView Policy Server Using Radia

Radia Policy Server Guide

Software Version: 4.0

for the UNIX and Windows operating systems

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

Trademark Notices

Linux is a registered trademark of Linus Torvalds.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Acknowledgements

PREBOOT EXECUTION ENVIRONMENT (PXE) SERVER
Copyright © 1996-1999 Intel Corporation.

TFTP SERVER
Copyright © 1983, 1993
The Regents of the University of California.

OpenLDAP
Copyright 1999-2001 The OpenLDAP Foundation, Redwood City, California, USA.
Portions Copyright © 1992-1996 Regents of the University of Michigan.

OpenSSL License
Copyright © 1998-2001 The OpenSSLProject.

Original SSLeay License
Copyright © 1995-1998 Eric Young (eay@cryptsoft.com)

DHTML Calendar
Copyright Mihai Bazon, 2002, 2003

Technical Support

Please select Support & Services from the following web site:
<http://www.hp.com/managementsoftware/services>

There you will find contact information and details about the products, services, and support that HP OpenView offers.

The support site includes:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information

About this Guide

Who this Guide is for

This guide is for Radia systems administrators who want to use the Radia Policy Server for administrative purposes such as mapping services to users in the directory tree. The Radia Policy Server saves time because you do not have to maintain lists of users in the Radia Configuration Server Database. Instead, directory services handle policy management, and Radia manages services.

What this Guide is about

The *Radia Policy Server Guide* discusses:

■ Installing the Radia Policy Server.

■ Configuring the Radia Policy Server for Lightweight Directory Authentication Protocol (LDAP) support.

■ Administering policy with the Radia Policy Server.

This guide documents the Radia Policy Server for Windows and UNIX version 3.2.

Summary of Changes

This printing of the *Radia Policy Server Guide* contains the following changes to information and procedures for the following chapters.

Chapter 3:
Configuring and Administering LDAP

■ Page 42, *Adding Radia Policy Attributes*: This section discusses what attributes need to be added to your directory service for use with the Radia Policy Server.

■ Page 46, *Support for Multiple LDAP Connections*: This feature allows for multiple synchronous LDAP connections for the Radia Policy Server.

■ Page 47, *Specifying the Radia Configuration Server*: This is a new section that describes how to specify the location of the Radia Configuration Server to the Radia Policy Server and the location of the Radia Policy Server to the Radia Configuration Server.

■ Page 51, *To create the LDAP method in the Radia Database*: When creating the LDAP_RESOLVE instance, the ZMTHTYPE must be set to REXX, not EXE as previously documented.

[3.2] Page 59, *Distributed Administration*: Radia Policy Server will now honor the access rights you assigned in your directory server to your Radia Policy Server administrators.

■ Page 59, *To enable distributed administration*: When using the SETACCESS option, make the change in the pm.cfg file, not the httpd.rc file as previously documented.

■ Page 60, *Configuring the Service Drop Down*: This section was renamed from Administering Policies in the previous version of this guide.

■ Page 60, *Configuring the Service Drop Down*: You can now add policy entitlements using a drop-down box in the Radia Policy Server's administrative interface.

■ Page 61, *Adding a Policy (EdmPolicy)*: New caution — if the Radia Policy Server is used to store services, do not connect the same ZSERVICE instance using both the Radia Database and the Radia Policy Server.

■ Page 70, Adding a Link (EdmLink): This section describes how to use the edmLink attribute to create a connection to a group that is not part of the user's LDAP group membership.

■ Page 75, *Policy Scope*: This new section illustrates the scope of policies set with Radia Policy Server.

[3.2] Page 78, *Controlling Policy Scope Globally:* This new feature allows you to control the scope of policy using the Radia Policy Server configuration file.

Appendix B:
Use Existing LDAP Attributes

3.2   Page 91, *Use Existing LDAP Attributes*: This new feature allows you to use an existing LDAP attribute for interaction with Radia Policy Server. This feature should only be used if you are not able to change your directory service schema.

Editorial Improvements

In addition to the changes listed above, this version contains various editorial, organizational, and style updates to each chapter and section and the index.

Conventions

You should be aware of the following conventions used in this book.

Table P.1 ~ Styles

Element	Style	Example
References	*Italic*	See the *Publishing Applications and Content* chapter in this book.
Dialog boxes and windows	**Bold**	The **Radia System Explorer Security Information** dialog box opens.
Code	`Andale Mono`	`radia_am.exe`
Selections	**Bold**	Click the **Next** button.

Table P.2 ~ Usage

Element	Style	Example
Drives (system, mapped, CD)	Italicized placeholder	*SystemDrive*:\Program Files\Novadigm might refer to C:\Program Files\Novadigm on your computer.
		CDDrive:\client\radia_am.exe might refer to D:\client\radia_am.exe on your computer.
Files (in the Radia Database)	All uppercase	PRIMARY
Domains (in the Radia Database)	All uppercase	PRIMARY.SOFTWARE
		May also be referred to as the SOFTWARE domain in the PRIMARY file.
Classes (in the Radia Database)	All uppercase	PRIMARY.SOFTWARE.ZSERVICE
		May also be referred to as the ZSERVICE class in the SOFTWARE domain in the PRIMARY file.

The table below describes terms that may be used interchangeably throughout this book.

Table P.3 ~ Terminology[*]

* Depends on the context. May not always be able to substitute.

Term	May also be called
Application	software, service
Client	Radia Application Manager and/or Radia Software Manager
Computer	workstation, server
NOVADIGM domain	PRDMAINT domain **Note**: As of the 4.0 release of the database, the NOVADIGM domain is being renamed the PRDMAINT domain. Therefore, if you are using an earlier version, you will see the NOVADIGM domain in the database.
Radia Configuration Server	Manager, Active Component Server
Radia Configuration Server Database	Radia Database

Contents

Contents

1

Introduction

At the end of this chapter, you will:

- Know the benefits of the Radia Policy Server.
- Understand Radia Policy Server Processing.

About Radia Policy Server

The Radia Policy Server is a Web server used for administration purposes such as mapping services to users in the directory tree. It is one of the Management Extensions in the Radia Infrastructure providing integration and extended enterprise functionality with your directory services. Policy method connections in the Radia Configuration Server Database are used to determine what services should be distributed and managed for the user that is currently logged on by querying the Radia Policy Server.

The Radia Integration Server service, installed with the Radia Policy Server, is a run-time technology that integrates HP infrastructure services. Radia Policy leverages your investment in directory services while using Radia for software management. This greatly reduces the total cost of ownership of your environment. In other words, directory services handle policy management and Radia manages services. This saves you time because you don't have to define or maintain lists of users in the Radia Configuration Server.

> **Note**
>
> The Radia Policy Server was formerly known as the Policy Manager. As of this printing, the name still remains Policy Manager in some of the configuration windows.

The Radia Policy Server integrates with Lightweight Directory Access Protocol (LDAP) directory servers and SQL databases to enable single source points of control for user authentication, access policies, and subscriber entitlement. These LDAP directory servers include Microsoft Active Directory, Novell NDS, and other vendor's LDAP servers, as well as Microsoft NT Domain Manager, Computer Associates ACF2 and Top Secret, and Oracle, Sybase and Microsoft SQL-based databases.

Benefits

Our goal is to provide the best policy-based management based upon the latest technologies. The HP vision of the Radia Policy Server can be summarized in the following points:

- **Simplicity**
  The model should be no more complex than your policies.
- **Sophistication**
  The model should be capable of expressing even the most subtle or complex policies you need.
- **Clarity**
  Each organizational policy should exist only once in the model, associated directly with the logical object that is the subject of that policy.
- **Investment Protection**
  The model should build upon your existing Directory Services infrastructure.
- **Openness**
  The model should be flexible.

Radia Policy Server Processing

The Radia Policy Server acts as a bridge between the Radia Configuration Server and a directory server. It is a separate component from the Radia Configuration Server. Therefore, when a customer has multiple Radia Configuration Servers, he may have a single Radia Policy Server co-located with his directory server. The following figure provides an overview of Radia Policy Server Processing.
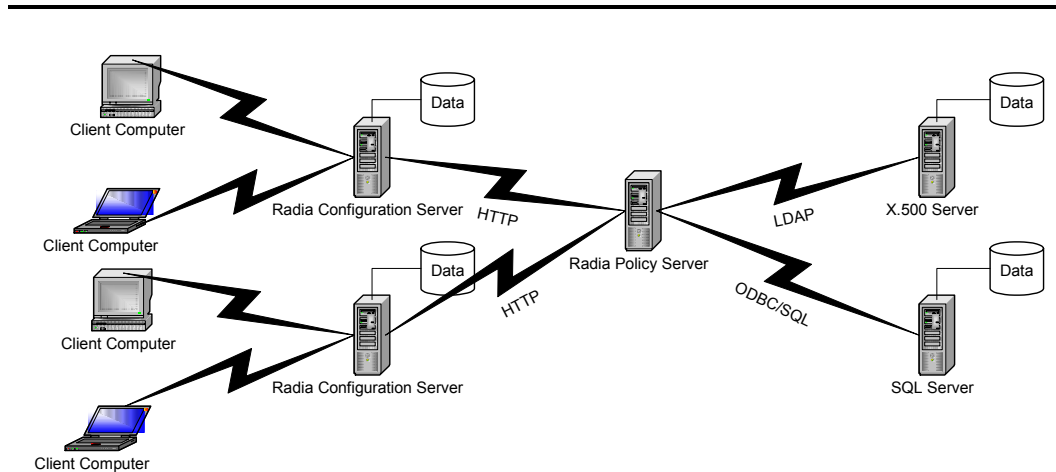

Figure 1.1 ~ Radia Policy Server Processing.

In Figure 1.1 above, the following things occur:

1. The Radia Client connects to the Radia Configuration Server to resolve its *desired state*. The desired state embodies the content that Radia manages for a specific client computer. The desired state for each client computer is dynamically created by the Radia Configuration Server based on information in the Radia Database. The desired state may also be described as the *distribution model*.

2. The Radia Configuration Server contacts the Radia Policy Server to perform policy resolution, and builds the client's desired state using the policy information.

3. The policy method, LDAP_RESOLVE, handles the resolution requests, converting the requests into HTTP queries to the Radia Policy Server, and treats the results as a set of objects and attributes to be incorporated into the desired state of the connected client.

4. The Radia Configuration Server completes resolution of the desired state and returns the information to the Radia Client.

The Radia Policy Server maintains a persistent connection to an LDAP directory server (or SQL database) and responds to policy requests by performing a policy resolution against the policy

database and returns the set of objects resolved as the result set of the HTTP query. It also provides an HTML-based interface for administering policy information.

The Radia Policy Server can also take input via HTTP POST from multiple Radia Configuration Servers. The Radia Policy Server can reformat data and insert or update associated database tables. It is also possible to provide HTML-based interfaces for generating operational reports on the current or historical activity of the system.

About the Radia Integration Server

The Radia Integration Server integrates independent modules, such as the Radia Inventory Manager, the Radia Management Portal, the Radia Mobile Management Server, the Radia Proxy Server, and the Radia Policy Server, giving them access to all the functions and resources under the control of the Radia Integration Server. The Radia Integration Server is *not* a separately installed product. Each module resides in the Radia Integration Server's modules directory. These Radia components use the same core Radia Integration Server files, and run under the same process.

Benefits of the Radia Integration Server are:

- All the products using the Radia Integration Server for Windows are loaded from a single NT Service called "Radia Integration Server".

- When the Radia Integration Server starts, it will scan its configuration file and try to load all the products marked as loadable.

- Each product is separately licensed.

- The Radia Integration Server provides Web services that are shared by all loaded modules, resulting in a single entry point for all HTTP (Web-based) requests. This integration provides performance, efficiency, and ease of maintenance in an adaptable and cohesive (server) framework.

The Radia Management Portal, Radia Inventory Manager, and Radia Proxy Server are described in the *Radia Management Portal Guide*, the *Radia Inventory Manager Guide*, and the *Radia Proxy Server Guide*, respectively.

About this Guide

In addition to this chapter, this book contains the following information:

- **Radia Policy Server Installation**
  This chapter describes how to install the Radia Policy Server.

- **Configuring and Administering LDAP**
  This chapter describes how to configure and administer your Radia environment with LDAP services.

Summary

■ The Radia Policy Server integrates with Lightweight Directory Access Protocol (LDAP) directory servers and SQL databases to enable single source points of control for user authentication, access policies, and subscriber entitlement.

■ The Radia Policy Server acts as a bridge between the Radia Configuration Server and a directory server.

■ The Radia Policy Server is a module of the Radia Integration Server.

2

Installation

At the end of this chapter, you will:

- Know how to install the Radia Policy Server.
- Be able to verify installation of the Radia Policy Server.

Note
This document covers installation information for Win32 servers only. Full product documentation is available on the HP Technical Support Web site.

Radia Policy Server Installation

Before you install the Radia Policy Server, identify the server where the Radia Policy Server will reside. Administrators usually choose the same physical server that is running the Directory Services (or SQL database), or the Radia Configuration Server. Review the reference documentation on the HP Technical Support Web site to help you determine which machine is best suited in your environment for running the Radia Policy Server. Install the Radia Policy Server from the Management Extensions directory on the Radia Infrastructure CD-ROM.

License File and Support

Before starting the installation, download your license file from the HP ftp site. This license file must be accessible to install the products that your enterprise purchased.

If you need assistance, contact HP Technical Support (see page 4).

Tips

■ Have the license file easily accessible for your installation.

■ Click **Cancel** in any of the windows to exit the installation. If you click **Cancel** accidentally, prompts enable you to return to the installation program.

■ Click **Back** at any time to return to previous windows. All the information that you entered thus far will remain unchanged.

■ Most windows have associated error messages. If your specifications are invalid, an error message will appear. Click **OK** and enter the correct information.

■ This installation program will display default values. We strongly recommend accepting all defaults; however, they can be overridden by specifying the parameters necessary to suit your environment.

Note

The Radia Management Portal, the Radia Mobile Management Server, the Radia Proxy Server, the Radia Policy Server, and the Radia Inventory Manager are each composed of modules that reside in the `Integration Server\modules` directory. When installed on Win32 platforms, previous versions of these Radia components each installed a service in this directory that reflected its own name. However, starting with the common infrastructure installation format of Radia 3.0, these Radia components use the same core Integration Server files, `nvdkit` and `httpd.tkd`, and run under the same process. Therefore, when you install one of these Radia 3.0 products, if another version of the Integration Server exists (on the server you are installing the product onto), the more recent version will prevail.

Platform Coverage

The LDAP Policy Extension is available on the following platforms:

■ Windows 2000, Windows NT 4.0 SP3 and above

■ Solaris 5.7

■ HP-UX 10.2

■ AIX 4.2

For additional information on the Radia Policy Server on other platforms, check the HP Technical Support Web site.

To install the Radia Policy Server for Windows

1. From the Radia Infrastructure CD-ROM, navigate to the `ManagementExtensions` `\PolicyServer` directory. Open the folder for your operating system.

2. Double-click **setup**.

The **Radia Policy Server Install** window opens.
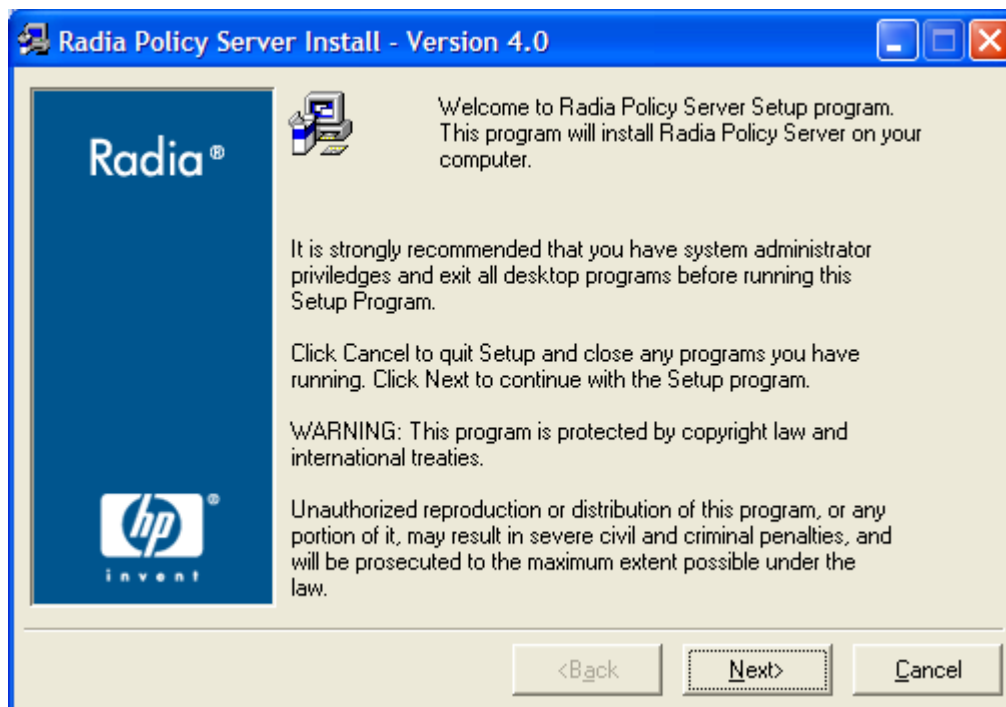
Figure 2.1 ~ Radia Policy Server Install Welcome window.

3. Click **Next**.

The License Agreement window opens.
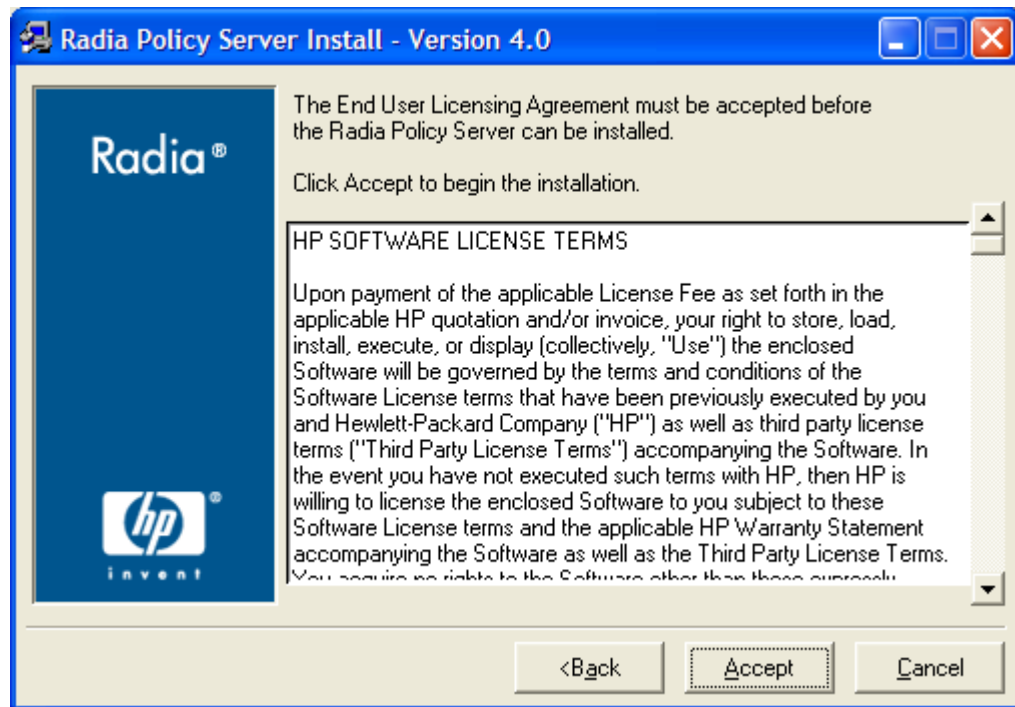
Figure 2.2 ~ Read the license agreement.

4. Read the license agreement and click **Accept**.

The **Select the install**a**tion folder** window opens.
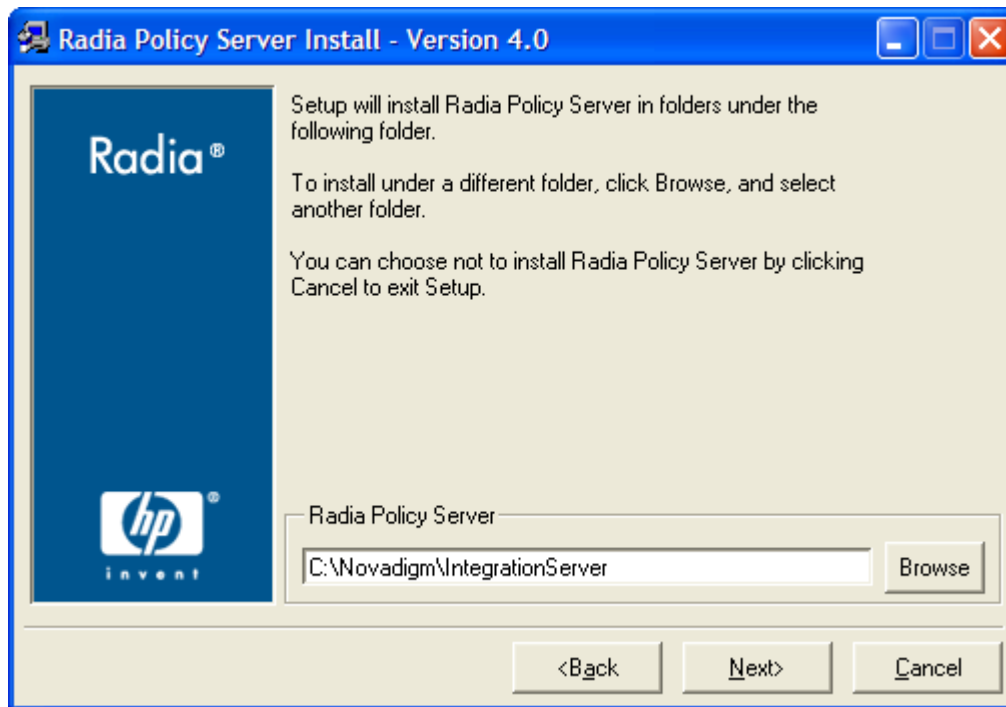
Figure 2.3 ~ Select the installation folder.

5. Use this window to select the folder where you want to install the Radia Policy Server.
   - Click **Next** to accept the default installation folder.
     OR
   - Click **Browse** to select a different folder.

6. Click **Next**.

The **Select License File** window opens.
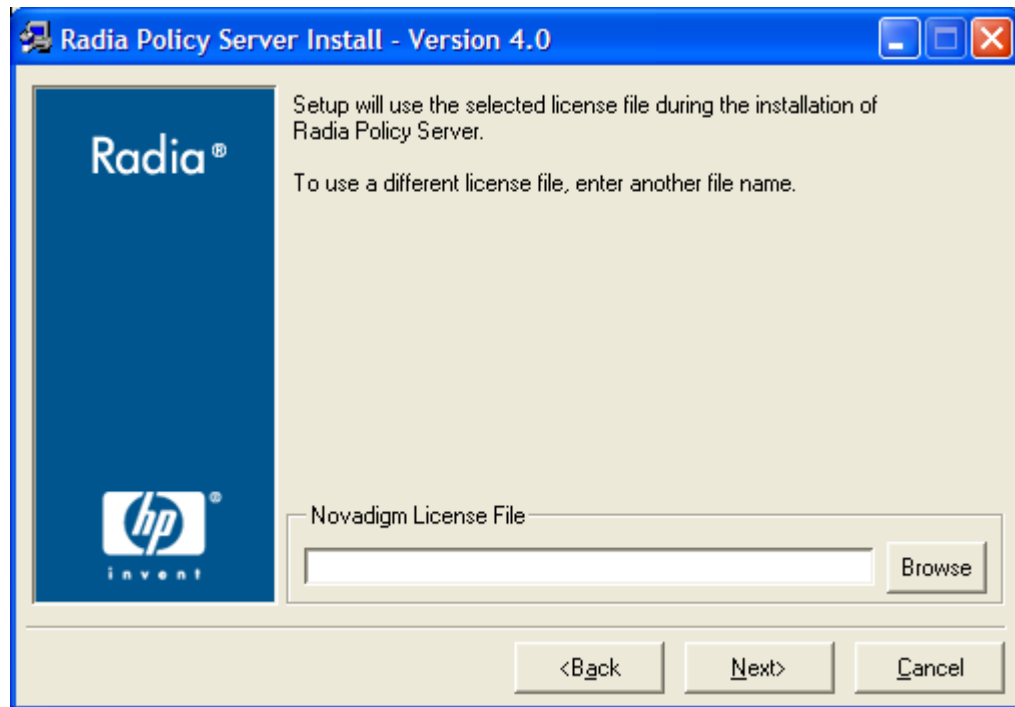
Figure 2.4 ~ Select the license file.

7. Click **Browse** to navigate to the location of your `license.nvd` file, and click **Open**. You will return to the **License Information** window, and the complete path to your license file will be displayed.

8. Click **Next**.

A summary of the installation information opens.
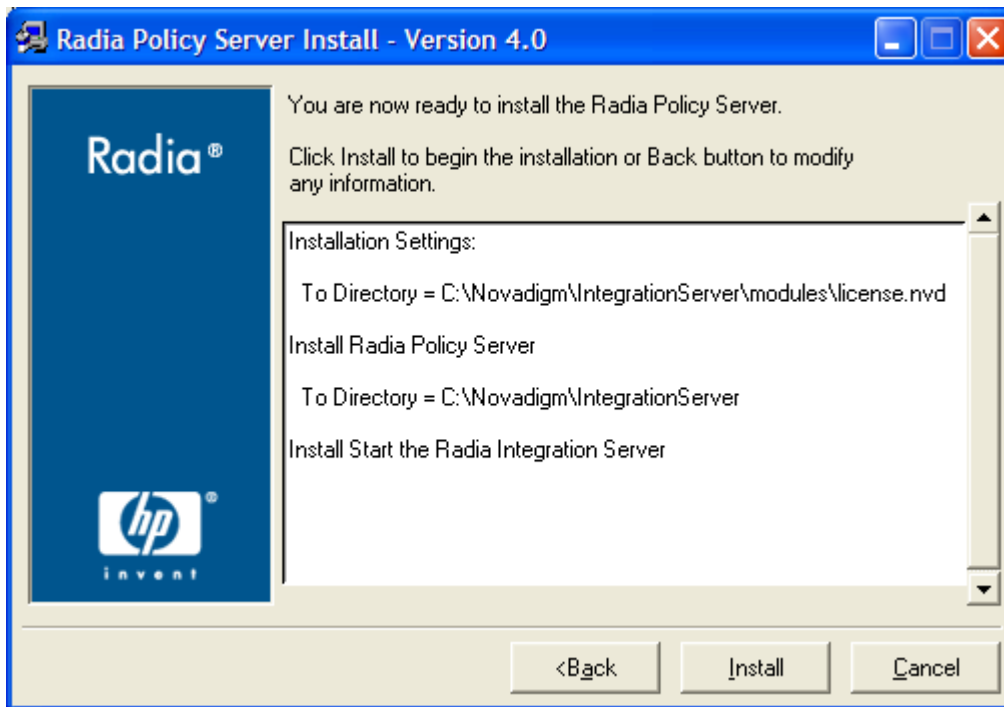
Figure 2.5 ~ Summary of installation information.

9. Click **Install** to begin the installation.

The installation progress window opens.
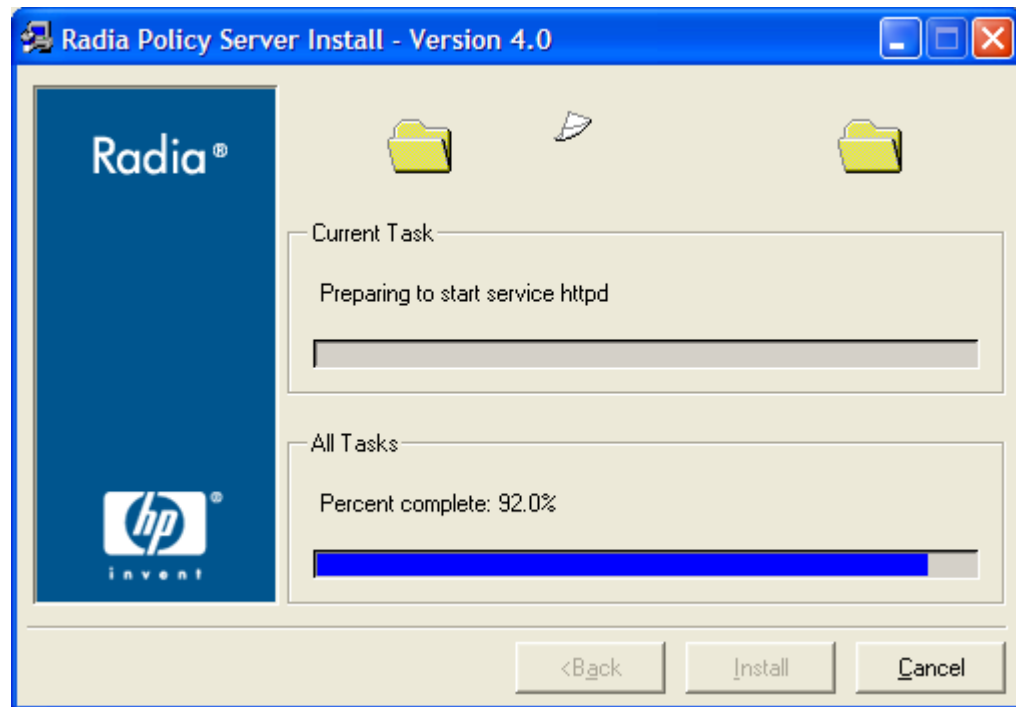
Figure 2.6 ~ Installation progress.

10. Click **Finish** when the installation is finished.
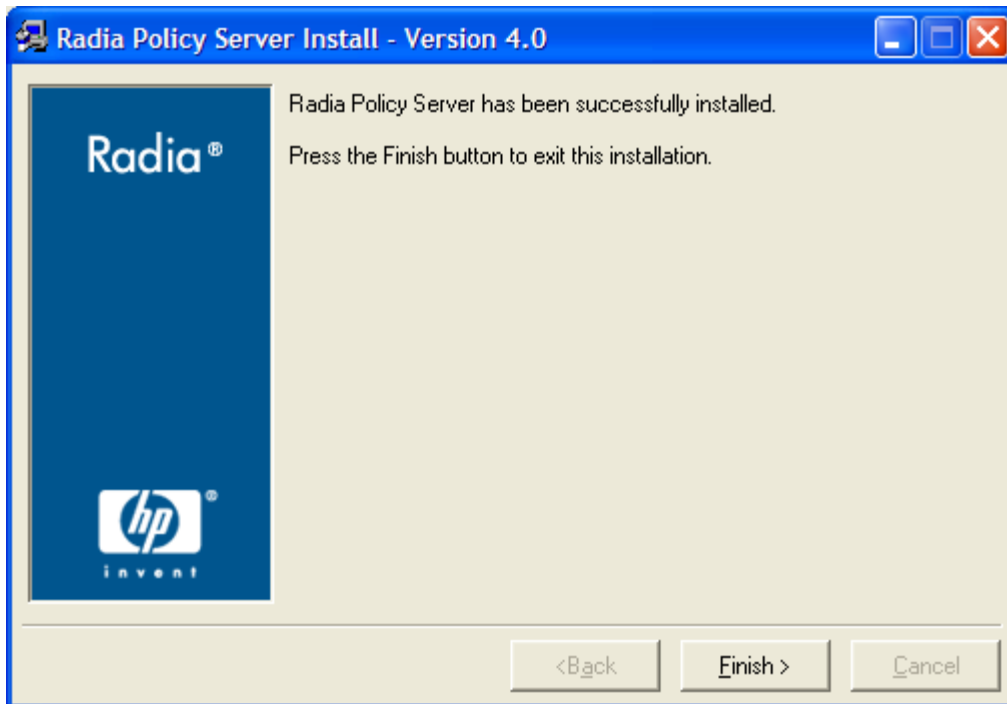
Figure 2.7 ~ Installation is finished.

Verify Installation

Confirm that the Radia Policy Server is running by performing the following verifications. If you are not using the Radia Management Portal, you can directly access the Radia Policy Server by following the procedure *To access the Radia Policy Server without the Radia Management Portal* below. You can administer the Radia Policy Server from the Radia Management Portal, if they are both installed on the same workstation. To do this, see the procedure *To access the Radia Policy Server if using the Radia Management Portal* on page 34.

Note

The Radia Policy Server was formerly known as the Radia Policy Manager. As of this printing, the name still remains Policy Manager in some of the configuration windows.

To access the Radia Policy Server without the Radia Management Portal

1. Open your Web browser.

2. In the Address bar, type `http://IP_Address:3466`.

The `IP_Address` is the IP address of the computer where the Radia Policy Server is installed.
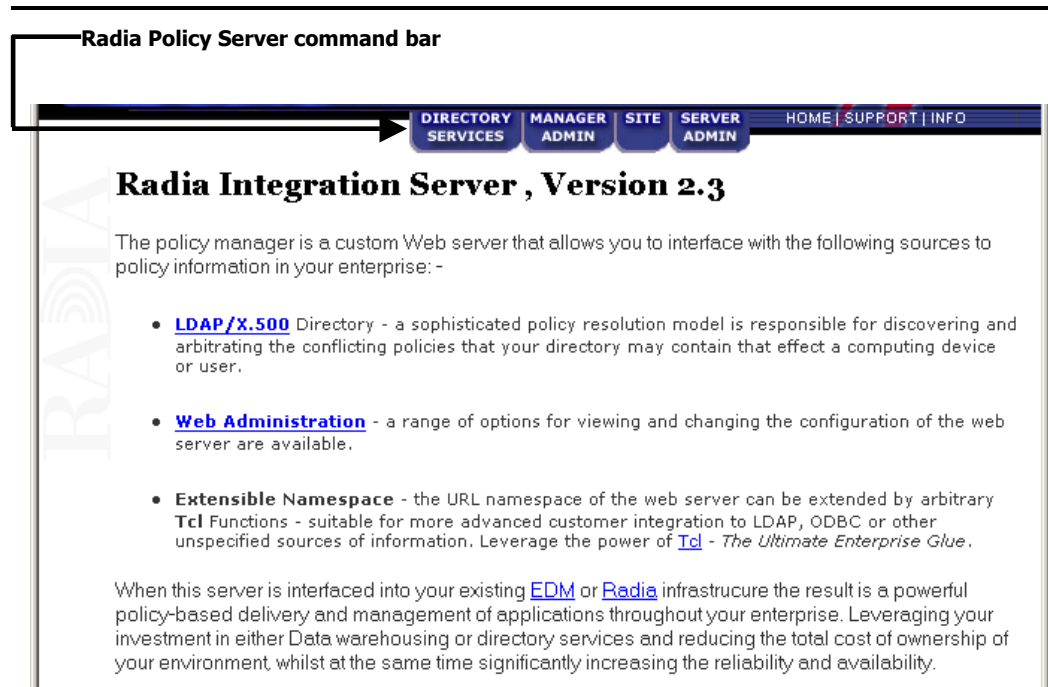
The Radia Integration Server Policy Server Web page opens.

Radia Policy Server command bar

Figure 2.8 ~ Radia Integration Server Policy Server site.

3. Click **Directory Services** in the command bar.

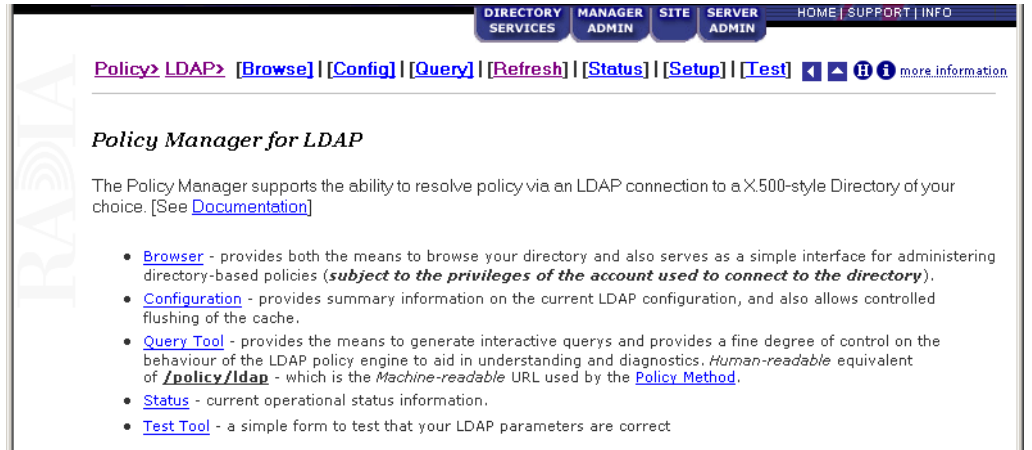The **Policy Manager for LDAP** page opens.

Figure 2.9 ~ Policy Manager for LDAP page.

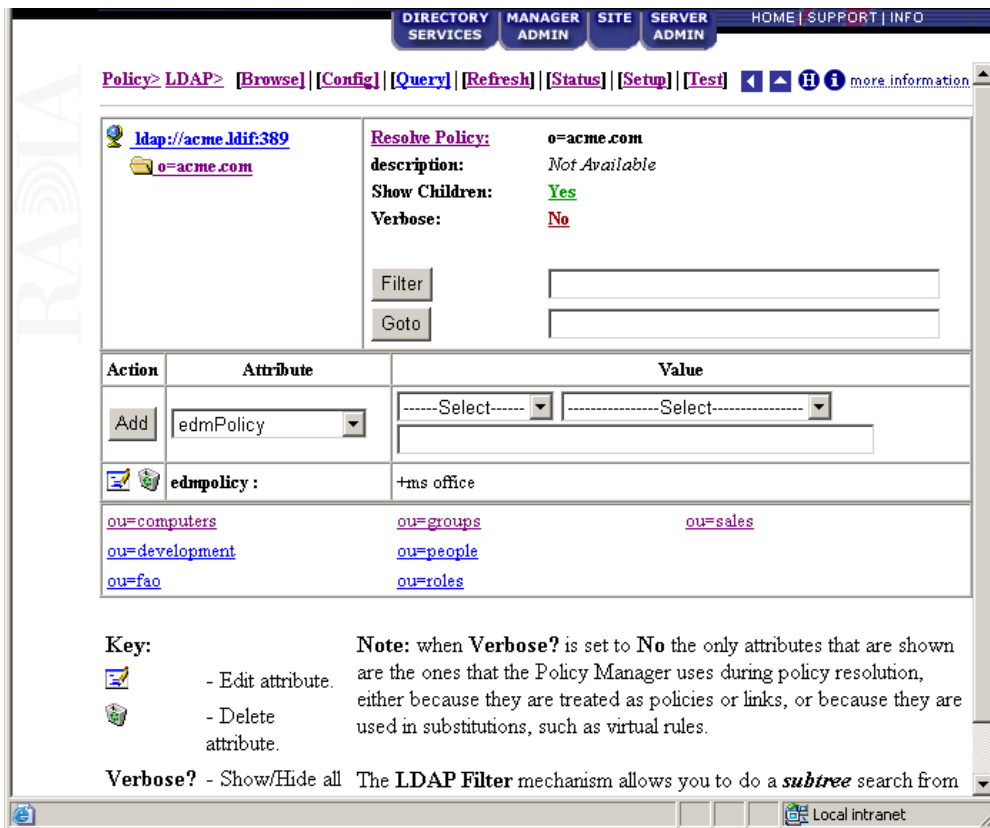**4.** Click **Browse** to verify that you can move through the sample database.

Figure 2.10 ~ Browse window for LDAP.

5. If you are on the same machine as the Radia Configuration Server, in the Radia Policy Server command bar click **MANAGER ADMIN** to verify that you can move through the Radia Database.

Figure 2.11 ~ Browse the Radia Database.

To access the Radia Policy Server if using the Radia Management Portal

1. Open your Web browser.

2. In the Address bar, type `http://IP_Address:3466`, and click **Go**.

The *IP_Address* is the IP address of the computer where the Radia Management Portal is installed.
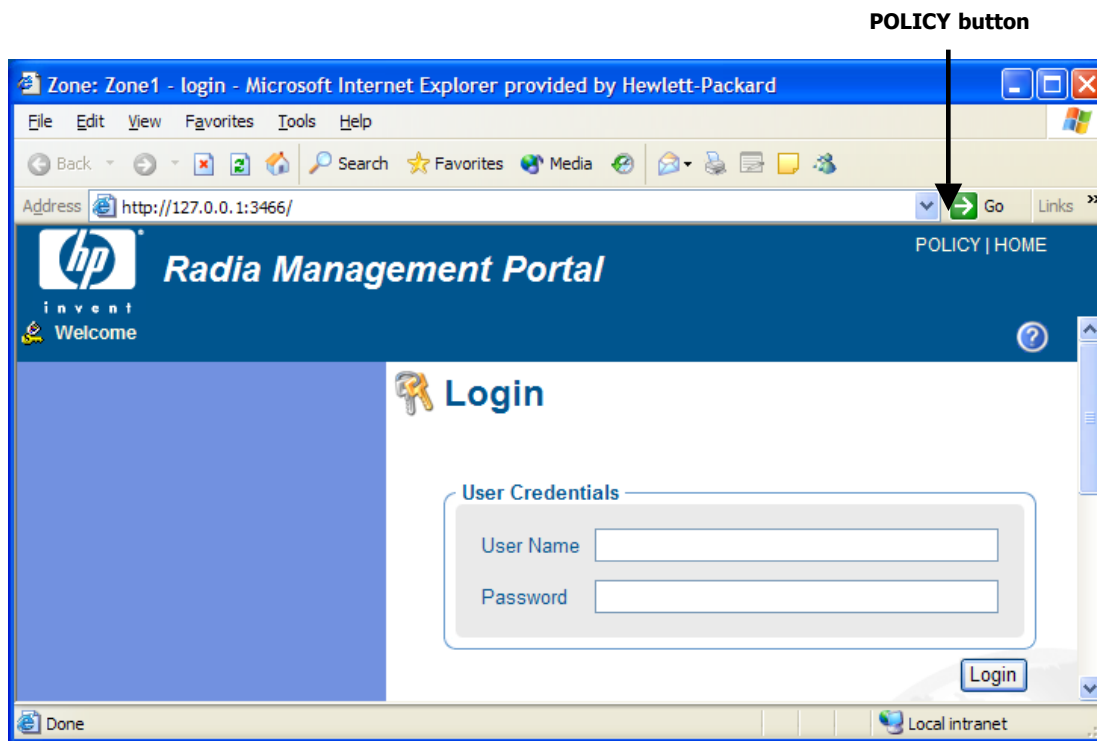
The Radia Management Portal opens.

POLICY button

Figure 2.12 ~ Management Portal Login window.

3.   Click **POLICY** in the banner of the Radia Management Portal.

> **Note**
>
> If the Radia Inventory Manager or Radia Policy Server is installed on the same computer as the Radia Management Portal, links will be available in the banner of the Management Portal.
>
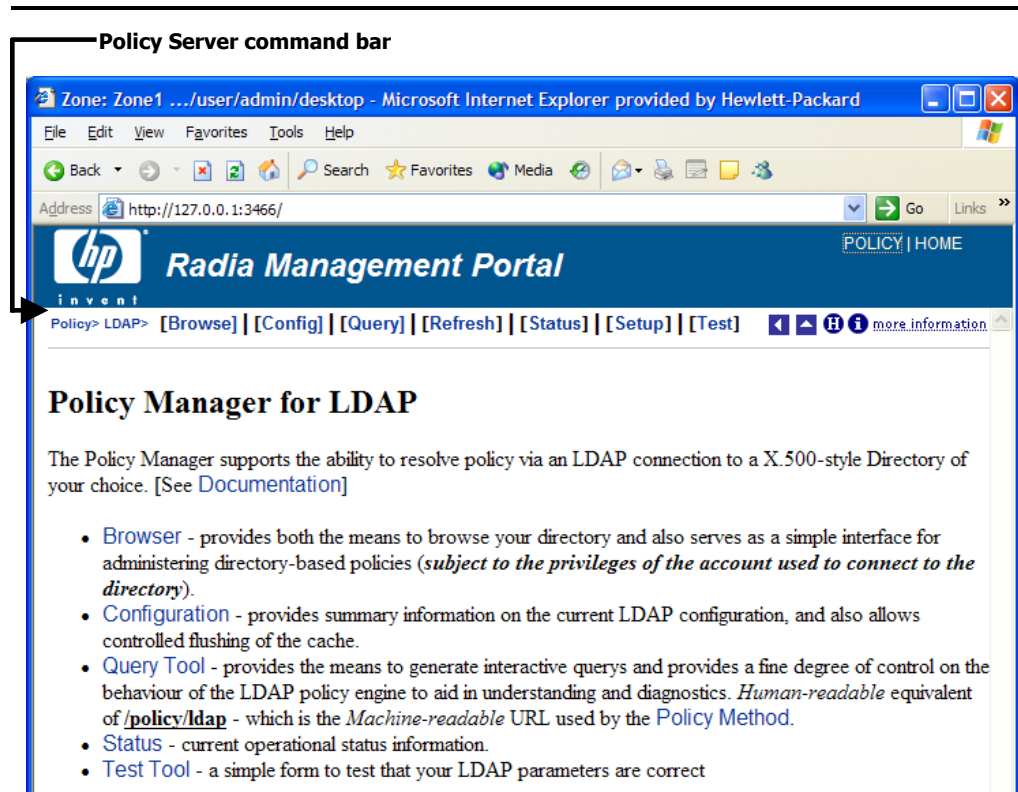> For example, in the figure above, click **POLICY** to access the Radia Policy Server.

Policy Server command bar

Figure 2.13 ~ Radia Policy Server Web site.

After installing the Radia Policy Server, you will use this site to navigate to the Lightweight Directory Access Protocol (LDAP) tree and assign services to users.

4. In the Radia Policy Server command bar, click **Browse** to verify that you can move through the sample database.
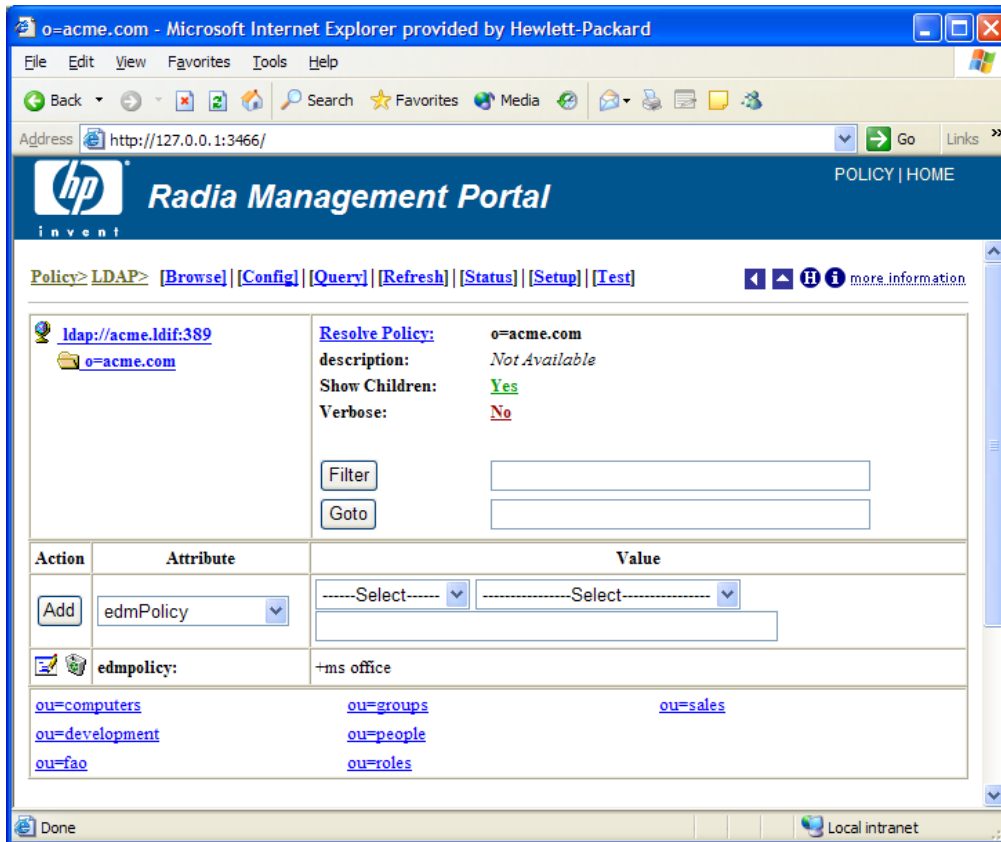
Figure 2.14 ~ Browse window for LDAP from the Radia Management Portal.

5. If you are on the same machine as the Radia Configuration Server, in the Radia Policy Server command bar click **MANAGER ADMIN** to verify that you can move through the Radia Database.

Figure 2.15 ~ Browse the Radia Configuration Server Database.

The Radia Policy Server is running.

Summary

■ Have the appropriate files ready before installing the Radia Policy Server.

■ Back up your Radia Database before installing the Radia Policy Server.

■ Verify installation using an Internet browser.

Configuring and Administering Policy

At the end of this chapter, you will:

■ Know what attributes to add to your directory service for use with Radia Policy Server.

■ Understand how to configure the method for resolving LDAP policies in the Radia Database.

■ Be able to administer policy.

You must complete the following steps to connect your directory services, the Radia Policy Server, and the Radia Configuration Server.

1. Add the required attributes for Radia Policy Server to your directory service.

2. Configure the Radia Policy Server to connect to your directory server.

3. If your Radia Configuration Server and Radia Policy Server are on two separate computers, you will need to configure them to communicate.

4. Configure the LDAP resolve method on your Radia Configuration Server to use your directory server.

5. Connect the LDAP resolve method to a policy instance in your Radia Database.

After completing these steps, you can begin to administer policy using the Radia Policy Server.

Adding Radia Policy Attributes

The Radia Policy Server requires that the LDAP schema of an existing directory implementation be modified before it can be used to manage policy. The schema changes are required in order to define three attributes: `edmFlags`, `edmLink`, and `edmPolicy`. These attributes are used to manage policy scope, relationships, and assignments. Consult your directory service documentation and your enterprise's directory service administrator to make these changes. Be sure to back up your directory scheme before any modifications.

Caution

Changes to the LDAP schema can be risky because modifications to many directory services are *not* reversible. Be sure you type correctly. Check and double check the values you are entering before saving the changes to each value entered into the directory schema. Consult your directory services administrator and documentation.

Add the following attributes:

■   Add `edmFlags` as a single-valued, integer attribute with an object ID of 1.3.6.1.4.1.2133.2.1.1. It controls the scope of your policy. This is an optional attribute.

■   Add `edmLink` as a multi-valued, case-sensitive string with an object ID of 1.3.6.1.4.1.2133.2.1.2. This attribute allows you to create a connection to a group that is not part of the user's LDAP group membership. This is an optional attribute.

■   Add `edmPolicy` as a multi-valued, case-sensitive string with an object ID of 1.3.6.1.4.1.2133.2.1.3. Use `edmPolicy` to assign services to users and groups. This is an optional attribute.

■   Add `edmPolicyOverride` as a multi-valued, case-exact string with an object ID of 1.3.6.1.4.1.2133.2.1.4. Use `edmPolicyOverride` to define policy overrides. This is an optional attribute.

■   Add `edmPolicyDefault` as a multi-valued, case-exact string with an object ID of
1.3.6.1.4.1.2133.2.1.5. Use `edmPolicyDefault` to assign policy defaults. This is an optional
attribute.

Adding the nvdObject Class

Some directory services, such as Microsoft Active Directory, do not allow adding of attributes to
the *top* class. This is the highest level in the schema. If you cannot add attributes to the *top* class,
create a class that will hold the `edmLink`, `edmFlags`, and `edmPolicy` attributes, and inherit the
values included in the *top* class. By creating this class, including its inherited values, we can
modify the areas needed to apply Radia policies to specific areas of the directory tree. If you can
add the attributes to the *top* class, policies can be placed anywhere in the tree.

If you need to create a class, name the class `nvdObject`. Create it as an auxiliary class with *top* as
its parent class. Set the object ID to 1.3.6.1.4.1.2133.2.1. After creating the `nvdObject` class, add
the `edmFlags`, `edmLink`, and `edmPolicy`. To proceed, you must reload your directory schema.
Consult your directory service's documentation for instructions on how to do this.

Modifying classes with nvdObject

Once the schema has been re-loaded, the values entered above will show up as a selection, and
you can add the `nvdObject` class to areas of your directory affected by the Radia Policy Server.

To complete the modification for Microsoft Active Directory, `nvdObject` must be added as an
Auxiliary class on the **Relationships** tab to all of the Active Directory classes listed below.

■   Person

■   Container

■   DomainDNS

■   Organizational Unit

■   Group

You have now completed the necessary modifications to your directory schema. See *To configure
the Radia Policy Server for LDAP* on page 44 for instructions on how to connect Radia Policy
Server to your directory services.

Note
If you are not able to change the schema, you can use an attribute that already exists in the directory schema.
This feature should only be used when it is NOT possible to make the necessary changes to the schema. See *Appendix B: Use Existing LDAP Attribute* starting on page 91, for instructions on how to do this.

Connection to LDAP

The LDAP extension supports a range of options that are stored in the LDAP start up script. This script is located in the Radia Integration Server directory. HP recommends changing the LDAP configuration through the Radia Policy Server's Setup page to perform validation of user input.

> **Note**
>
> If you make manual changes to `pm.cfg`, you will need to restart the Radia Policy Server or Radia Integration Server service.

Below is a procedure for setting the LDAP configuration. Table 3.2 on page 46 describes all of the possible configuration values.

> **Note**
>
> For the ability to bind an Active Directory domain and edit Policy objects, the BIND_DN needs to have read access rights to the entire directory and write access rights to the top of the tree to which it will be editing.

To configure the Radia Policy Server for LDAP

1. Open either the Radia Management Portal or the Radia Integration Server. If you are using the Radia Management Portal, click on **POLICY** in the banner. From the Radia Policy Server page, click the **Setup** page.

Setup/Configuration

Any changes made here will effect the running service, and also be saved to disk.

Type	⦿ ldap ◯ ldif
Ldif	
Host*	10.10.10.12
Port*	389
Version	◯ 2 ⦿ 3
Base Dn	dc=asdfoods,dc=com
Bind Dn	cn=Administrator ,cn=Users,dc=asdfoods,dc=cor
Bind Pw	********
Prefix*	edm

Figure 3.1 ~ Connecting to a Microsoft Active Directory server.

2. For **Type**, select the **ldap** option.

3. In the **Base Dn** line, type the base domain. This is the highest level of the directory structure. If you leave it blank, the highest level is assumed.

Table 3.1 ~ BASE_DN and BIND_DN Examples

Item	Microsoft Active Directory	Novell Directory Service
Base Dn	Specifies the base domain. **Example**: dc=asdfoods, dc=com.	Specifies the base organization. **Example**: o=asdfoods
Bind Dn	Specifies the fully qualified name of the account that has Active Directory Schema Permissions on the Directory. **Example:** cn=Administrator , cn=Users, dc=asdfoods, dc=com **or** administrator@asdfoods.com	Specifies the fully qualified name of the account that has NDS Permissions on the Directory. **Example**: cn=Admin, ou=Users, o=asdfoods

4. In the **Bind Dn** line, type the fully qualified name of the account that has update authority to the specific OUs and containers to which the edmPolicy attributes will be applied. See Table 3.1 above for examples.

5. In the **Bind Pw** line, type in the password of the Account name referred to in the Bind Dn.

6. In the **Host** line, type the hostname or IP address of the Active Directory Server you wish to bind to for resolving policies.

7. Click **Submit** to submit the changes to the Radia Integration Server or Radia Policy Server service.

Table 3.2 ~ Configurable Values in the Web Interface		
Field	**Default**	**Description**
Host	localhost	Hostname or TCP/IP address of LDAP Server/Gateway.
Port	389	TCP/IP port of LDAP Server/Gateway.
Version	2	LDAP Protocol version to use (2 or 3)
Base Dn		DN of the logical root of the Directory—used to constrain the directory browser. Also used for $pinging$ the directory server periodically to ensure it is up.
Bind Dn		DN of account to use when authenticating (**BIND**) with directory. If this parameter is not supplied, then an $anonymous$ **BIND** is performed.
Bind Pw		Password for Bind Dn account. **Note**: This is stored in plain text in `pm.cfg`. It is *highly recommended* that customers secure access to the `<root>/etc` directory for administrators only.
CACHE	1	Enable caching (0 or 1).
DELAY	1	The delay in seconds between each retry attempt.
FLUSH_FREQ	3600	The delay in seconds between each flush of the cache.
RETRY	1	Number of attempts to issue the LDAP request before marking the directory as unavailable. If this occurs, a reconnection attempt will be made when the next ping is performed.
PING_FREQ	300	The delay between each attempt to search Base Dn (in seconds). This enables the Radia Policy Server to reconnect to a directory server than may have been restarted, and also serves as an active monitor of the availability of the directory.
TIMEOUT	120	Timeout (in seconds) for LDAP request.

Support for Multiple LDAP Connections

Radia Policy Server supports multiple concurrent LDAP queries. Configure the number of concurrent LDAP queries in the Radia Policy Server's configuration file, `pm.cfg`, The default location of this file is `<System Drive>:\Novadigm\IntegrationServer\etc`. Use a text editor such as Notepad to edit the file. The table below describes which parameters apply. When you make changes to `pm.cfg`, you will need to restart the Radia Integration Server service.

Table 3.3 ~ Configurable Values for Multiple LDAP Queries		
Value	**Default**	**Description**
N_WORKERS	4	Specifies number of parallel LDAP directory connections to be created.
PolicyUrl	/policy/ldap	Registers the URL of the Radia Policy Server's LDAP.
		This is required to use the N_WORKERS parameter. The parameter name is case sensitive. If you do not have this line in your pm.cfg, then you will need to add it.

Specifying the Radia Configuration Server

If your Radia Configuration Server is not on the same computer as your Radia Policy Server, you will need to specify the location of the Radia Configuration Server. To do this, edit the Radia Configuration Server's profile file, edmprof.dat, and the Radia Policy Server's configuration file, pm.cfg.

To specify the location of the Radia Policy Server on the Radia Configuration Server

1. On the Radia Configuration Server computer, open the Profile Editor. This opens the Radia Configuration Server's profile file, edmprof.dat, in a text editor.

2. Go to the [MGR_POLICY] section as shown in *Figure 3.2 ~ Edit the [MGR_POLICY] section of edmprof.dat* below.

```
* Manager Policy Section                                            *
*  HTTP_HOST      = Host name of Radia Policy Server                *
*                   Multiple hosts may be specified (space or comma *
*                   seperated) for fail over                        *
*  HTTP_PORT      = IP Port number of Radia Policy Server           *
*                   NO restart required                             *
*------------------------------------------------------------------*
[MGR_POLICY]
HTTP_HOST = XXX.XXX.XXX.XXX
HTTP_PORT = 3466
```

Figure 3.2 ~ Edit the [MGR_POLICY] section of edmprof.dat.

3. Type the IP address of the Radia Policy Server as the value for HTTP_HOST.

4. Type the port of the Radia Policy Server as the HTTP_PORT.

5. Save and close the EDMPROF.DAT.

After specifying to the Radia Configuration Server, where the Radia Policy Server is located, you need to specify to the Radia Policy Server where the Radia Configuration Server is.

To specify the location of the Radia Configuration Server to the Radia Policy Server

1. Open the Radia Policy Server's configuration file, `pm.cfg`, using a text editor. This file is located in the Integration Server's etc directory.

2. Type the IP address of your Radia Configuration Server as the value for the RCS_CACHE_HOST. If the port is different from the default of RCS_CACHE_PORT, change that value as well.

3. Save and close the modified `pm.cfg`.

4. Stop and restart the Radia Integration Server service.

Configuring the LDAP Method

If you are using LDAP, you must create a connection to the LDAP method in the Radia Database, and connect the users to the LDAP method. Perform the following two procedures to prepare your Radia Database to use the Radia Policy Server.

To create the LDAP method in the Radia Database

1. In the Radia System Explorer, go to SYSTEM.ZMETHOD.

2. Right-click **Methods (ZMETHOD)**.

   A shortcut menu opens.

3. From the shortcut menu, select **New Instance**.

   The **Create Instance** dialog box opens.
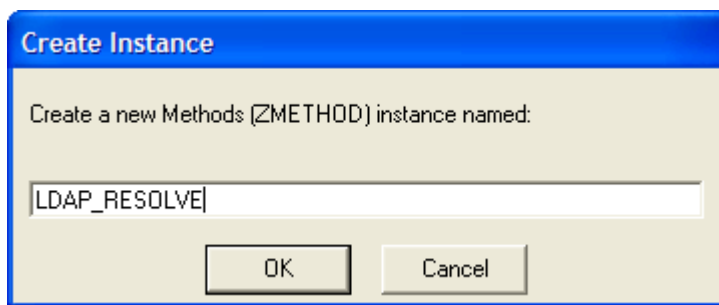
Figure 3.3 ~ Value for Utility Method.

4. Type **LDAP_RESOLVE** in the text box, and click **OK**.

   The **Radia System Explorer** window opens.

5. Double-click **LDAP_**.

   The tree expands.

6. Double-click **LDAP_RESOLVE** in the tree view.

The attributes of LDAP_RESOLVE appear in the list view.
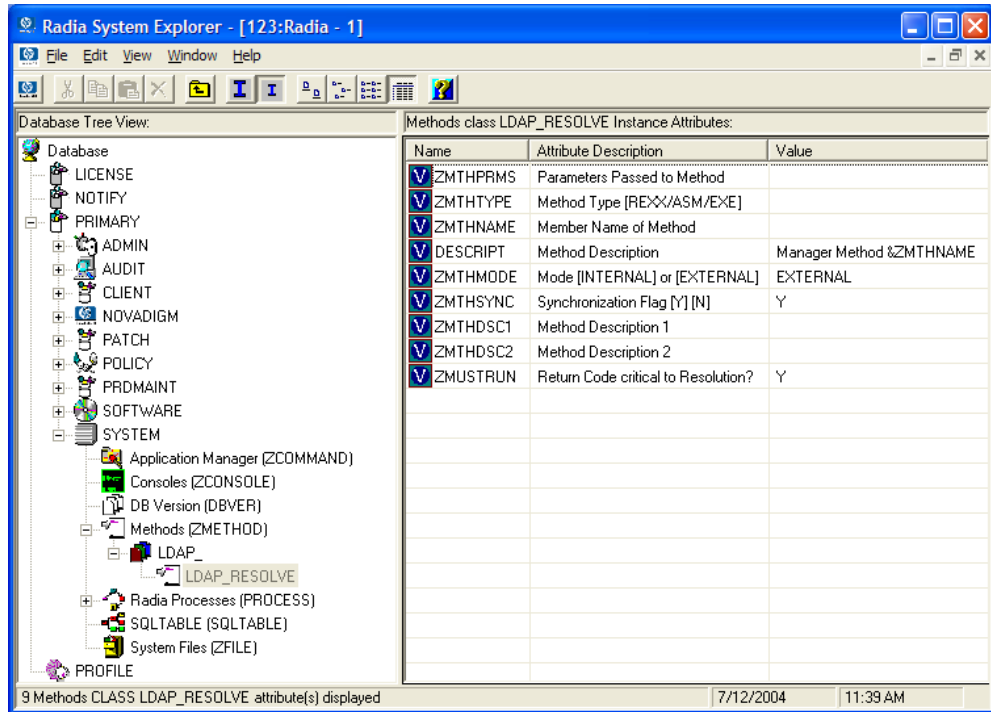
Figure 3.4 ~ LDAP_RESOLVE attributes.

7. Double-click the **ZMTHNAME** attribute in the list view.

The **Editing Instance** dialog box opens.

8. In the **Member Name of Method** field, type `radish`.

9. Click **ZMTHTYPE**.
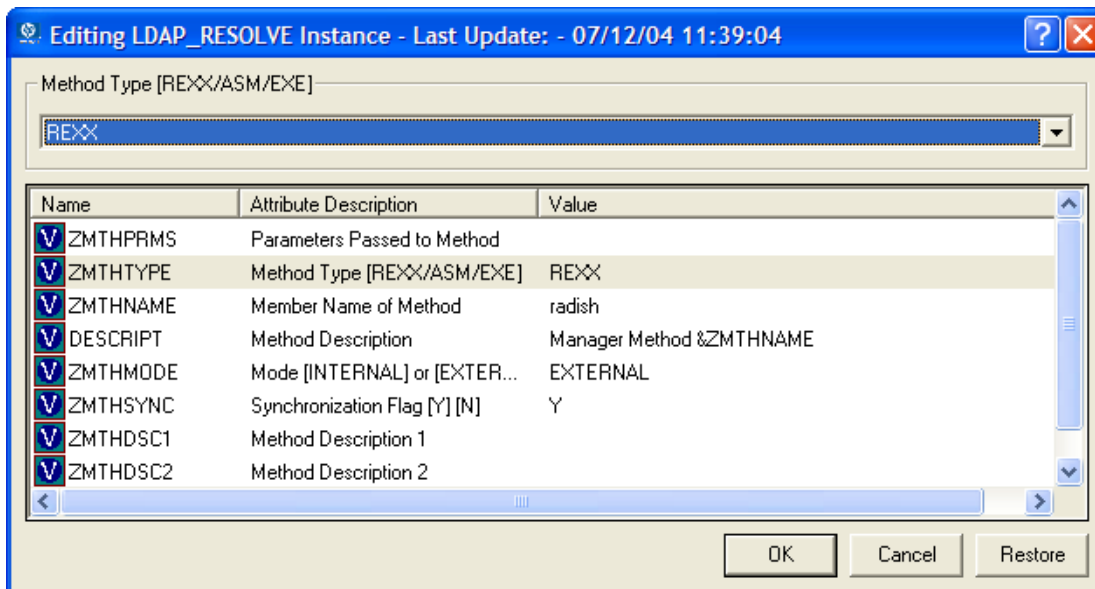
Figure 3.5 ~ Edit ZMTHTYPE attribute.

10. In the **Method Type** drop-down list, select **REXX**.

11. Click **ZMTHPRMS**.
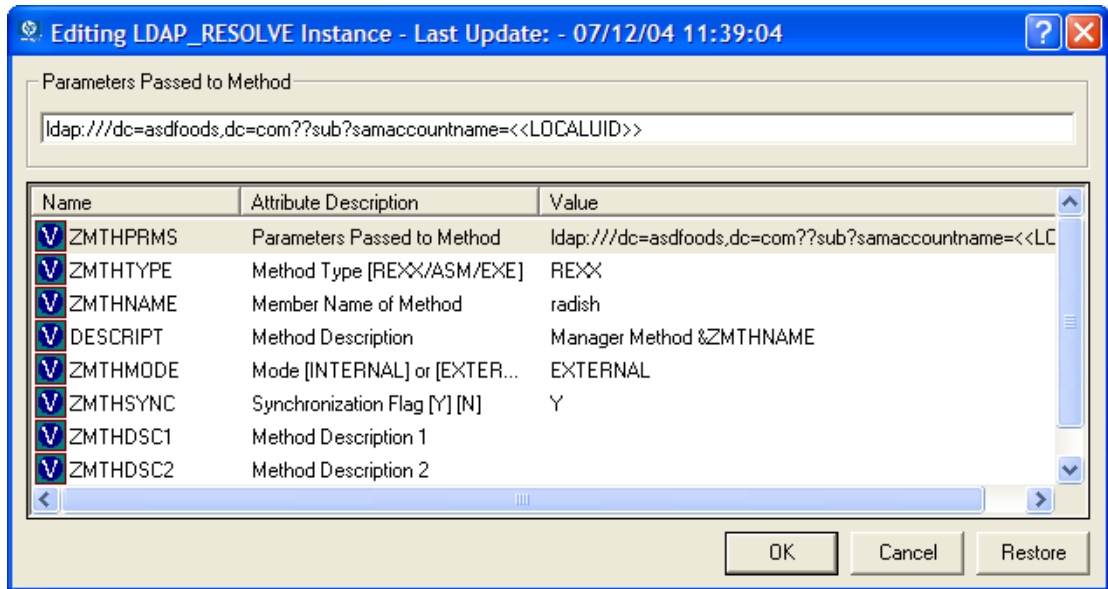
Figure 3.6 ~ Edit ZMTHPRMS attribute.

12. In the **Parameters Passed to Method** text box, use the following values.

For HTTP type:

`http://policy/ldap?dn=<<ZDN>>&&os=<<ZOS>>`

For Microsoft Active Directory:

- To manage policies by user type:

`ldap:///dc=domainname,dc=forestname,dc=com??sub?samaccountname=`
`<<LOCALUID>>`

For example,

`ldap:///dc=asdfoods,dc=com??sub?samaccountname=<<LOCALUID>>`

- To manage policies by machine type:

`ldap:///dc=domainname,dc=forestname,dc=com??sub?samaccountname=`
`<<ZUSERID>>$ (If the client uses $MACHINE as the ZUSERID)`

`http://policy/ldap?dn=<<COMPDN>>$`

For example,

`ldap:///dc=asdfoods,dc=com??sub?samaccountname=<< ZUSERID>>$`

For Novell Directory Services (NDS):

- To search the entire NDS tree for policy, type:

`ldap:///o=organization??sub?cn=<<ZNTUSER>>`

For example,

```
ldap:///o=cert??sub?cn=<<ZNTUSER>>
```

● To search NDS with a specified Distinguished Name, type:

```
http:///policy/ldap?dn=<<ZMASTER.DN>>
```

For example,

```
http:///policy/ldap?dn=<<ZMASTER.DN>>
```

For Netscape iPlanet:

● To manage policies by user type:

```
Ldap:///dc=com??sub?uid=<<ZUSERID>>
```

13. Click **OK**.

The **Instance Edit Confirmation** dialog box opens.

14. Click **Yes** to confirm the changes. The **Radia System Explorer** window opens.

Now, whenever a user logs onto Radia, the null instance calls the policy method, and will point to the appropriate services for that user.

Specifying the Distinguished Name

If there is no way to search the LDAP directory for a unique attribute, such as *samaccountname* in Active Directory, you will need to specify the distinguished name for each subscriber on each client computer (in the ZMASTER object). This must be done because there is no lookup from the Radia logon screen to the distinguished name in LDAP due to a limitation in LDAP.

To specify the distinguished name (dn)

1. Go to **Start, Programs, Radia Administrator**, and click **Radia Client Explorer**.

2. Go to **SystemDrive:\Program Files\Novadigm\Lib**.
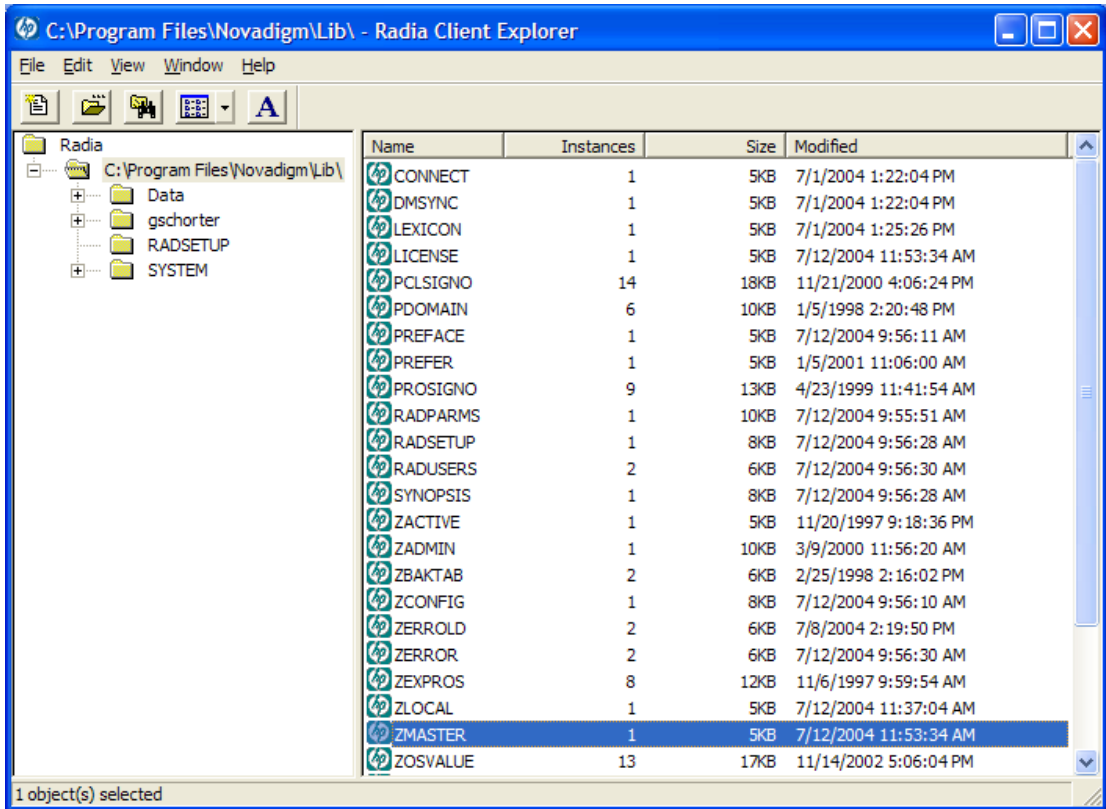
Figure 3.7 ~ Select local ZMASTER object.

3. Double-click the ZMASTER object.

4. From the **Variable** menu, click **Add**.

The **Add Variable to Object** dialog box opens.
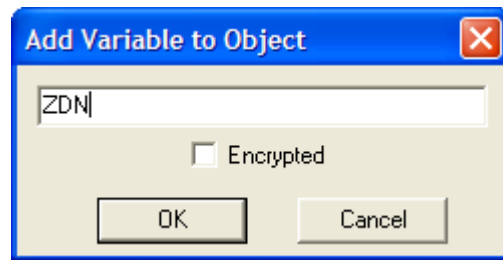
Figure 3.8 ~ Add a ZDN variable to the ZMASTER object.

5. In the text box, type a name for the variable, such as `ZDN`.

6. Click **OK**.

The **Change Variable** dialog box opens.

Figure 3.9 ~ Value for ZDN.

7. Enter the distinguished name information, such as `cn=gretchen schorter, ou=people, o=acme.com`.

8. Click **OK**.

9. Click **Save/Exit**.

Connecting to the LDAP Method

You must connect the LDAP method to an instance in the POLICY domain for policy resolution.

To connect the user to the LDAP method

1. Open the **Radia System Explorer**.

2. Navigate to **PRIMARY.POLICY.USER**.

3. Double-click the null instance.

> **Note**
>
> If the null instance is connected to the Default workgroup, change the name of the instance from Default to _NONE_.
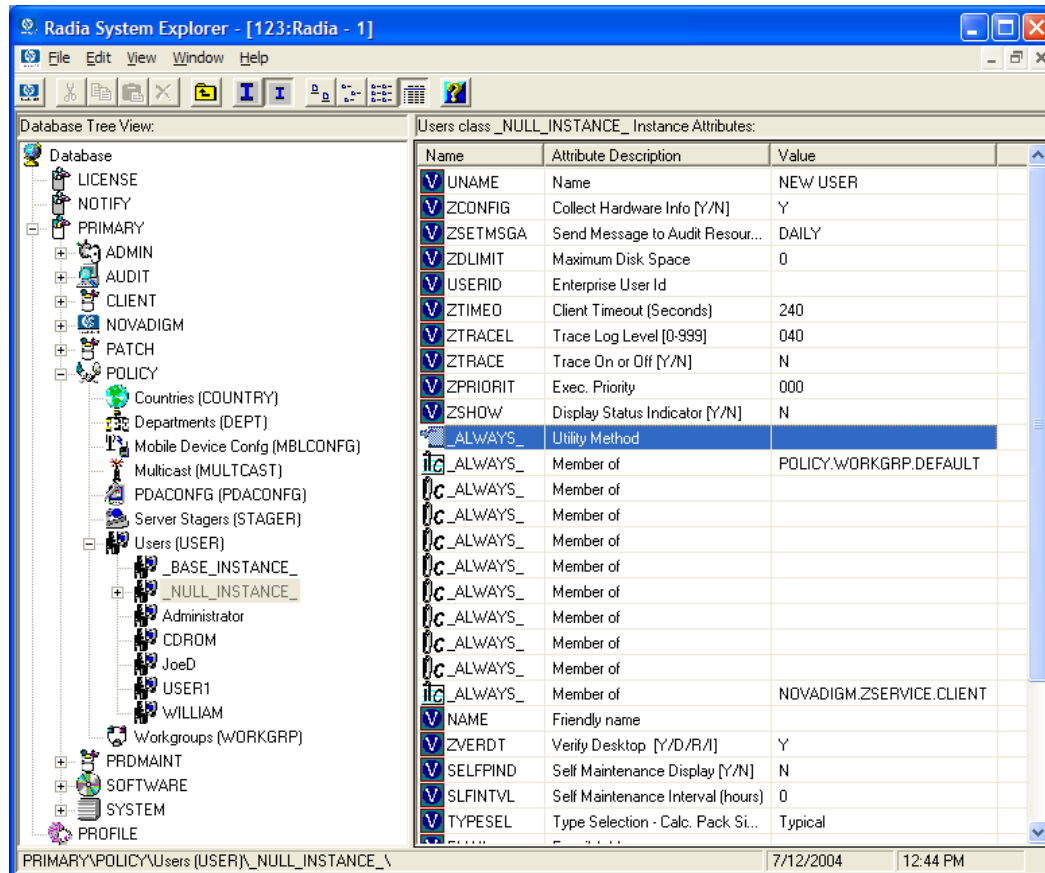
Figure 3.10 ~ Always Utility Method in the Radia System Explorer.

4. In the list view, double-click **_ALWAYS_**.

The **Editing Instance** dialog box opens.
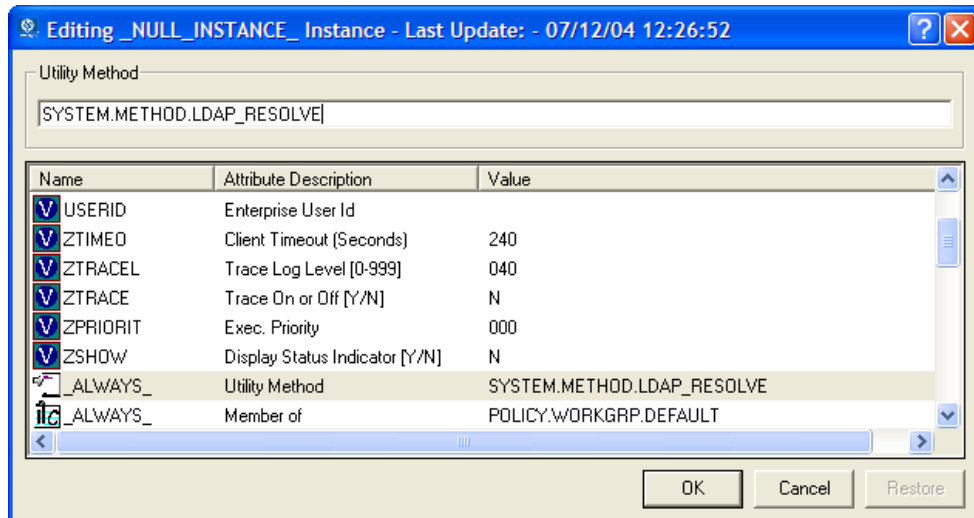
Figure 3.11 ~ Edit the null instance.

5. In the **Utility Method** text box, type `SYSTEM.ZMETHOD.LDAP_RESOLVE`.

6. Click **OK**.

The **Instance Edit Confirmation** opens.
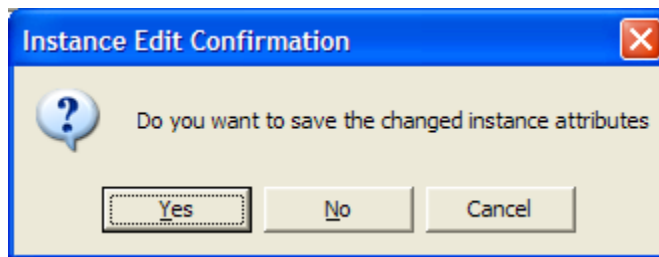
Figure 3.12 ~ Instance Edit Confirmation

7. Click **Yes**.

The LDAP_RESOLVE method is connected to the Null User instance.

Distributed Administration

By default, the Radia Policy Server connects to your directory service using the configured base DN. This user has administrative access to the entire directory starting from the configured base DN. In other words, all Radia Policy administrators have full access to the entire LDAP directory.

You can limit access for a Radia Policy administrator by defining his access in your LDAP directory. Then, when this administrator logs in, he will be prompted for a password, and will only have rights to the parts of the directory that you defined.

Figure 3.13 ~ Log in screen for Radia Policy Administrator

The administrator will be prompted to re-enter a user name and password if the connection has timed out. The logon times out when there is no activity on the Web page for a period of time. The default period is two minutes. To change the timeout, add the following to the pm.cfg file in the Radia Integration Server's etc directory:

```
ADMIN_TIMEOUT  <timeout in seconds>
```

To enable distributed administration

1. Use a text editor to open the pm.cfg file from the Radia Integration Server's etc directory.

2. Add or uncomment the following line from the pm.cfg:

   ```
   SetAccess /ldap "Policy Server" DIR <LDAP search URL>
   ```

3. Replace <LDAP search URL> with the appropriate location for your enterprise. It should be typed all in one line.

Example:

```
SetAccess /ldap "Policy Server" DIR
"ldap:///cn=users,dc=your_company_dc,dc=com??sub?(samaccountname=<<user>>)"
```

4. Save and close the file.

Distributed administration for your Radia Policy Server has been enabled. Set security for your Radia Policy administrator in your directory service.

Configuring the Service Drop Down

Now that you have configured your Radia environment to use the policies created in Radia Policy Server, you can begin administering it. To use the Radia Policy Server's drop-down box for the service list, you will need to create and configure a user who is entitled to all of the services you want to manage using the Radia Policy Server. If you are using Microsoft Internet Explorer version 6.0 or above, the services will be listed alphabetically under the domain where the service resides. Otherwise, the services will not be sub-divided by domain.

1. Use Radia System Explorer to create a user, such as POLUSER, in the Radia Database's POLICY domain.

2. Use the Radia System Explorer to assign POLUSER any Radia services that you want to manage with the Radia Policy Server.

> **Note**
>
> Keep your list of services to a manageable size. However, if you need to be able to manage all of the services using the Radia Policy Server, type **SOFTWARE.ZSERVICE.*** into one of the _ALWAYS_ connections for POLUSER. This will connect POLUSER to all Radia services.

3. Use a text editor to edit the `pm.cfg` file's RCS_CACHE_USER to be POLUSER. This connects the Radia Policy Server to the services for the POLUSER. The default location of `pm.cfg` is `<System Drive>`:\Novadigm\IntegrationServer\etc.

Adding a Policy (EdmPolicy)

The procedure *To add a policy* below shows an example of how to add a software entitlement attribute using the Radia Policy Server.

Caution

If Radia Policy Server is used to store services, do not connect the same Application (ZSERVICE) instance using both the Radia Database and the Radia Policy Server. In addition, only ZSERVICE instances should be assigned using Radia Policy Servers. The only notable exception to this rule is connecting an instance of the POLICY.STAGER class to a user or group of users in LDAP for the purposes of assigning a Radia Staging Server or Radia Proxy Server to a user or group of users. See KB01211 TECHNOTE: Assigning a Stager instance through the Policy Server on the HP Technical Support Web site for more information.

To add a policy

1. Open your Internet browser to either the Radia Management Portal page or the Radia Policy Server page.
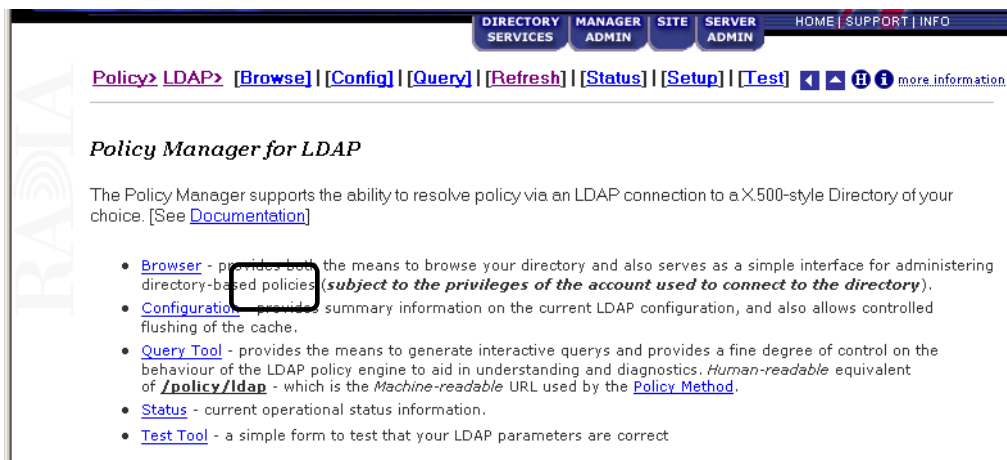
Figure 3.14 ~ Click Browse on the Radia Policy Server page.
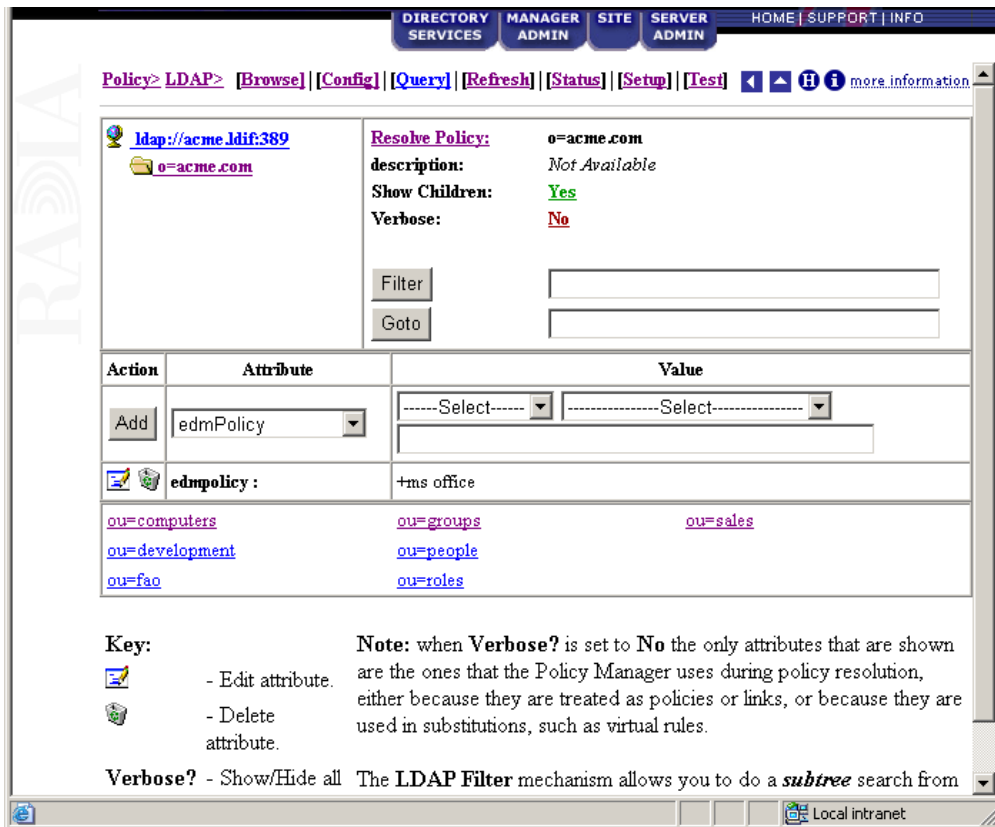
2. Click **Browse**.

Figure 3.15 ~ Click Browse to see your Directory Services.

3. Click on an organizational unit in the bottom section of the policy management screen to select it for policy management. In Figure 3.16 on page 63, we selected the **sales** organizational unit.
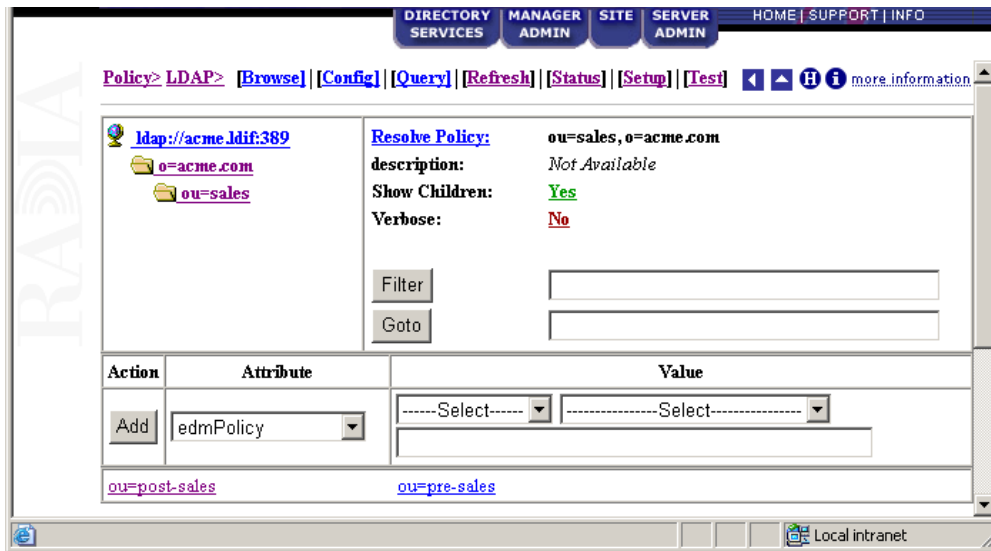
Figure 3.16 ~ ou=sales is selected for Policy management.

4. Make sure that **edmPolicy** is selected in the drop-down box for **Attribute**.

5. From the first drop-down box in the **Value** section, select the appropriate keyword. The options are *may*, *may not*, *should*, and *should not*. See *Table 3.4 ~ Assigning Services with the Pick List* on page 64 for additional information. After selecting the keyword, the value box will display the appropriate character for that value. In our example, we selected **May**.
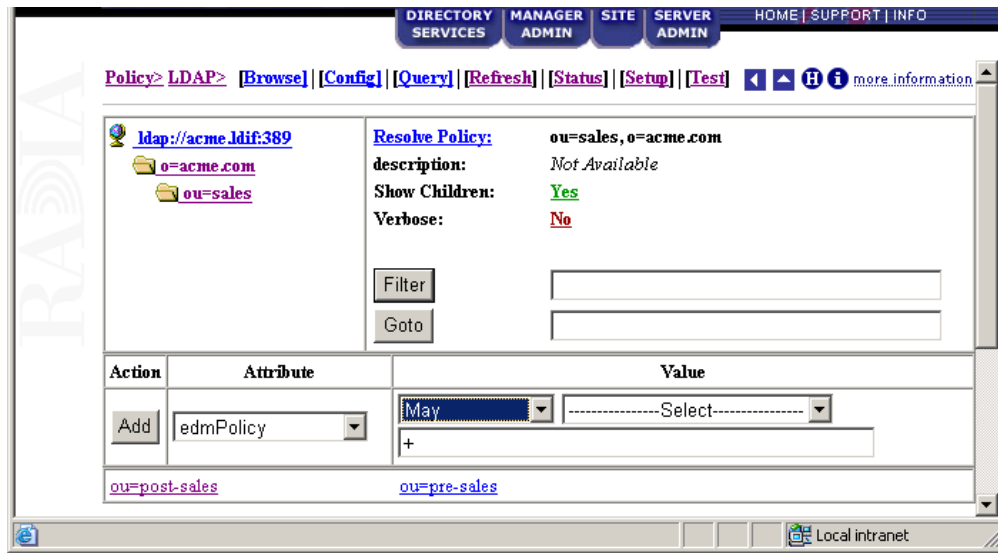
Figure 3.17 ~ Select May in the first drop-down box.

Table 3.4 ~ Assigning Services with the Pick List

If the value entry box, shows . .	You have specified . . .
+	that an application *may* be delivered.
-	that an application *may not* be delivered. This overrides +.
++	that an application *should* be delivered. This overrides -.
--	that an application *should not* be delivered. This overrides ++.

6. From the second drop-down box in the **Value** section, select the appropriate Radia Application (ZSERVICE) in its appropriate domain. In our example, we selected Amortize.
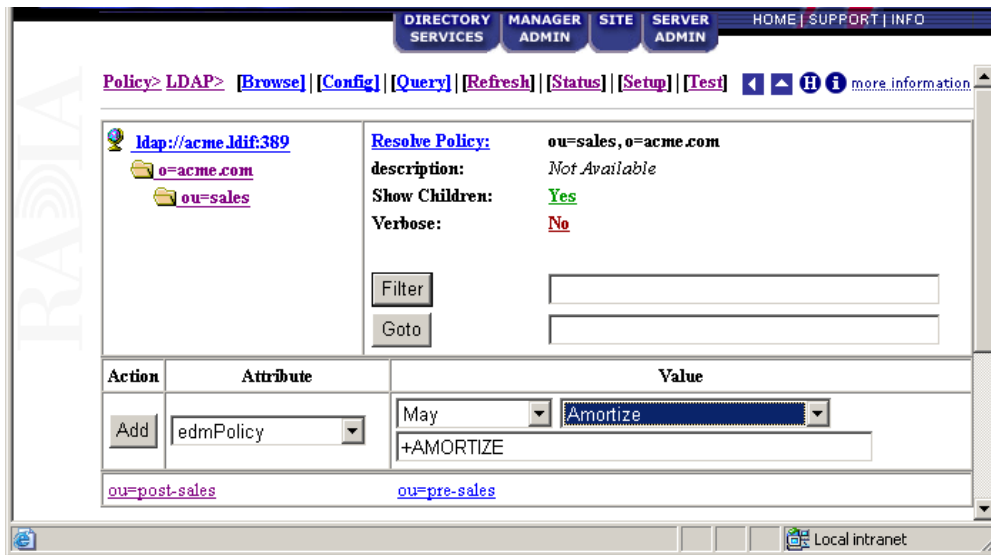
Figure 3.18 ~ The Sales container may receive Amortize.

7. Click **Add**. The Amortize service is added to the policy model.

> **Note**
>
> If you do not want to use the drop down box, you may type in your policy entitlement instead of using the values shown in *Table 3.4 ~ Assigning Services with the Pick List.*
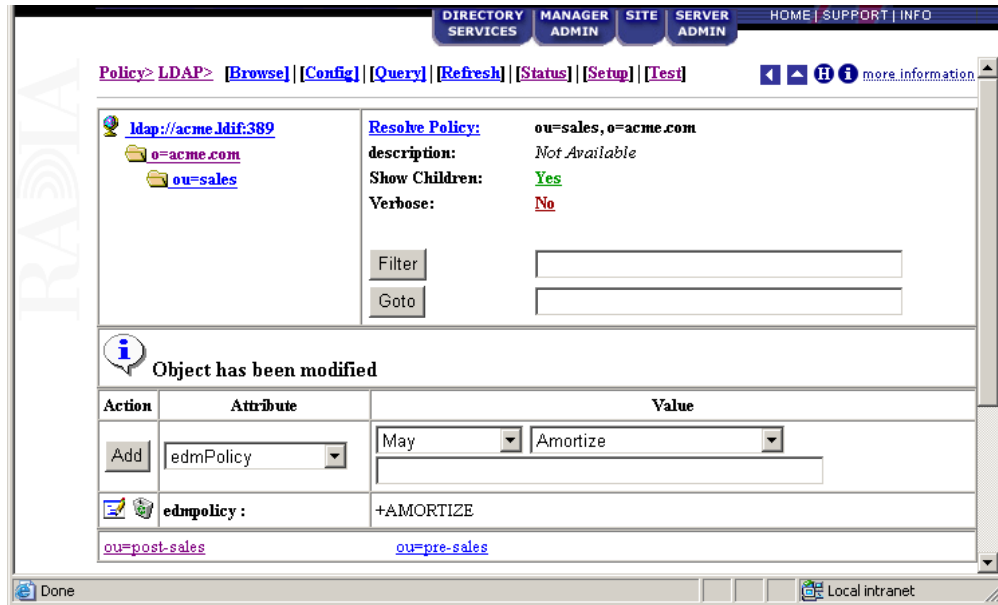
Figure 3.19 ~ Amortize is added to ou=sales.

Removing a Policy

As your software model continues to develop, you may want to remove a policy. Use the Radia Policy Server interface to remove policy.

To remove a policy

1. Open your Internet browser to either the Radia Management Portal page or the Radia Policy Server page. This defaults to port 3466 on your Radia Policy Server.

2. Click **Browse**.

3. Navigate to the group or user whose policy you want to modify.

4. Click on the recycle-bin icon 🗑 in the **edmPolicy** field that is associated with the application.
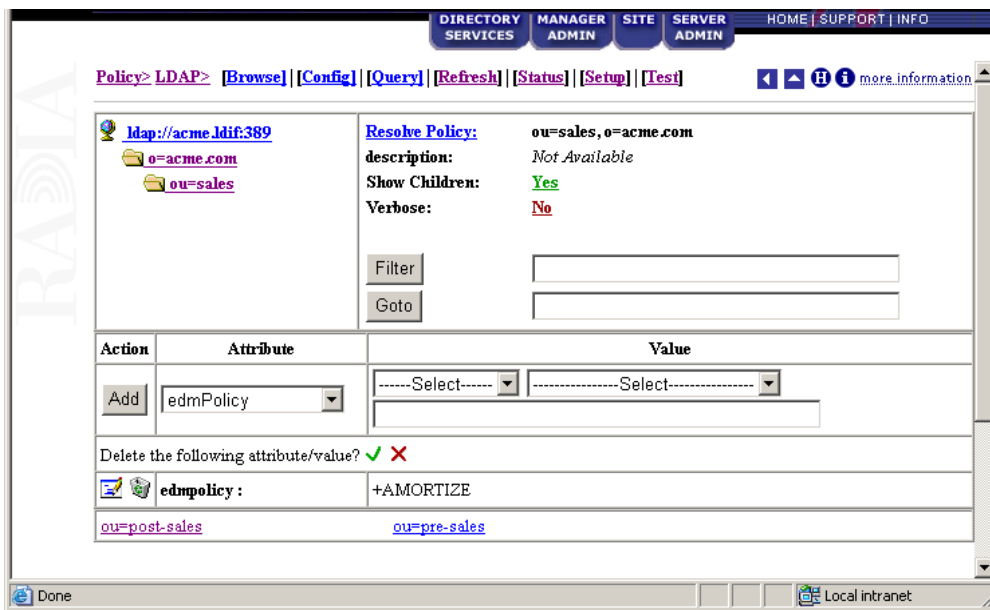
Figure 3.20 ~ Remove a policy.

5. Click on the ✔ icon to confirm the removal of this policy.
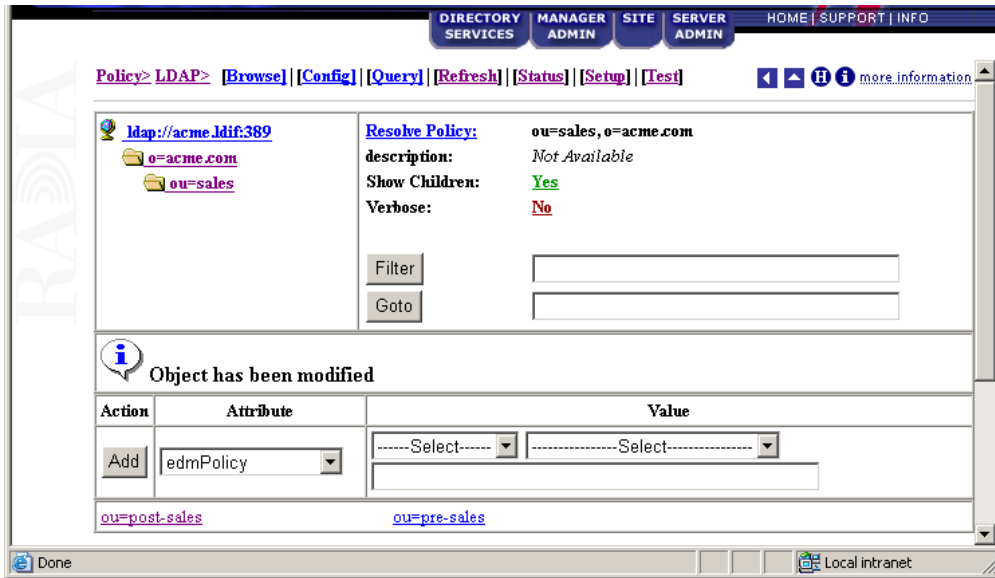
Figure 3.21 ~ Policy removed.

The policy for the Amortize service is removed.

Setting Policy Defaults and Overrides (EdmPolicyDefault and EdmPolicyOverride)

In addition to the existing values of the **edmpolicy** attribute, which either grant or deny access to an application (or object), two other types of policy are available.

■ **edmPolicyDefault** - this policy neither grants nor denies access - but *if* access has been granted, then these values are used as the **default template** for that policy. For example, this can be used to provide default attributes that typically appear in the <...> section of a policy - such that the result is identical to if they had actually been present in the policy.

> **Note**
>
> If attributes of the same name appear in an actual granting policy then those have priority.

■ **edmPolicyOverride** - this policy neither grants nor denies access, but *if* access has been granted, then these values **override** any equivalent attributes that were provided in the actual granting policy.

For a given application, more than one default may be encountered when resolving policy. In this case the defaults are ranked lowest to highest priority based upon the pri attribute only with lower numeric value being a higher priority. The same applies to for overrides.

The actual resulting policy that is returned to the Radia Configuration Server will be the logical set union performed as an ordered overlay. In other words, same named attributes are replaced. This will be performed as follows:

1. Lowest to Highest Priority DEFAULTS (0...n occurances)
2. Actual Granting Policy (always singular)
3. Lowest to Highest Priority OVERRIDES (0...n occurances)

Example 1 - simple override

- policy: Firefly <version=7 mode=typical>
- override: Firefly <version=8>
- **OUTCOME**: Firefly <version=8 mode=typical>

Example 2 - simple default

- policy: Firefly <mode=typical>
- default: Firefly <version=7>
- **OUTCOME**: Firefly <version=7 mode=typical>

Example 3 - default and override

- default: Firefly <mode=typical>
- policy: Firefly <version=7 issue=4>
- override: Firefly <version=8 mode=complete>
- **OUTCOME**: Firefly <version=8 issue=4 mode=complete>

Example 4 - mutiple defaults and multiple overrides

- default: Firefly <version=7> - NOTE: **pri** defaults to 10
- default Firefly <version=6 pri=5>
- policy: Firefly <mode=typical>
- override: Firefly <mode=complete> - NOTE: **pri** defaults to 10
- override: Firefly <mode=typical pri=5>
- **OUTCOME**: Firefly <version=6 mode=typical>

> **Note**
>
> Neither **defaults** nor **overrides** have any affect to policy resolutions that do *not* grant access to the subject (Firefly in the above example). Defaults and overrides only effect policy objects that are already granted access to an application - and the effect that they have is only to refine the definition of that access by possibly altering the set of attributes that contribute to the POLICY object that is present when the subject object is resolved on the Radia Configuration Server.

Adding a Link (EdmLink)

You may need to add a subscriber to a group without using your network's directory service. To do this, use the edmLink attribute. EdmLink allows you to create a connection to a group that is not part of the user's LDAP group membership. Then, the subscriber will inherit both the group membership assignments from LDAP and the assignments created with the edmLink attribute.

> **Caution**
>
> EdmLink should be used sparingly in the directory model. Its primary goal is to represent policy relationships between two objects that are not otherwise present in the form of parent-child or memberOf relationships

In the example below, we will add Albert Kirkman to the research organizational unit so that he will inherit any services assigned to that unit.

To add a link

1. Open your Internet browser to the Radia Policy Server page.
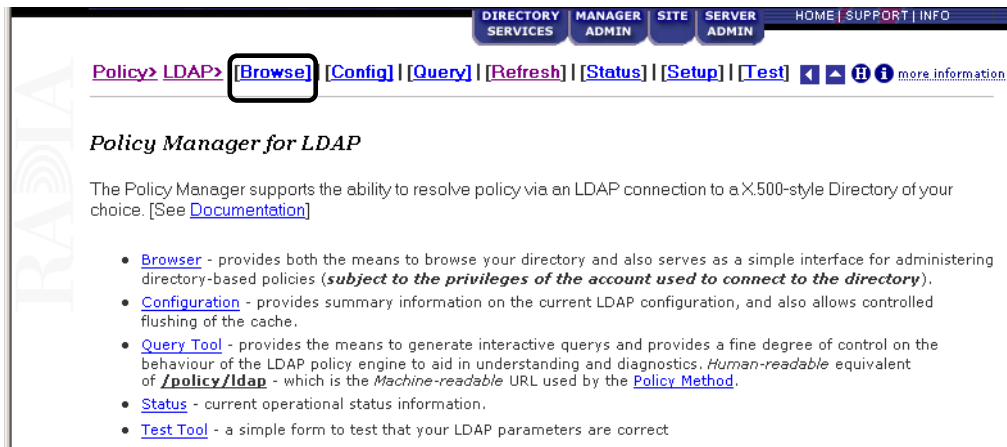
Figure 3.22 ~ Click Browse on the Radia Policy Server page.

2. Click **Browse**.

3. Click on an organizational unit in the bottom section of the policy management screen to select it, or continue to select until you reach a particular user. In *Figure 3.23 ~ cn=albert kirkman is selected for Policy management* on page 72, we selected **albert kirkman** from the **People** organizational unit in **acme.com**.
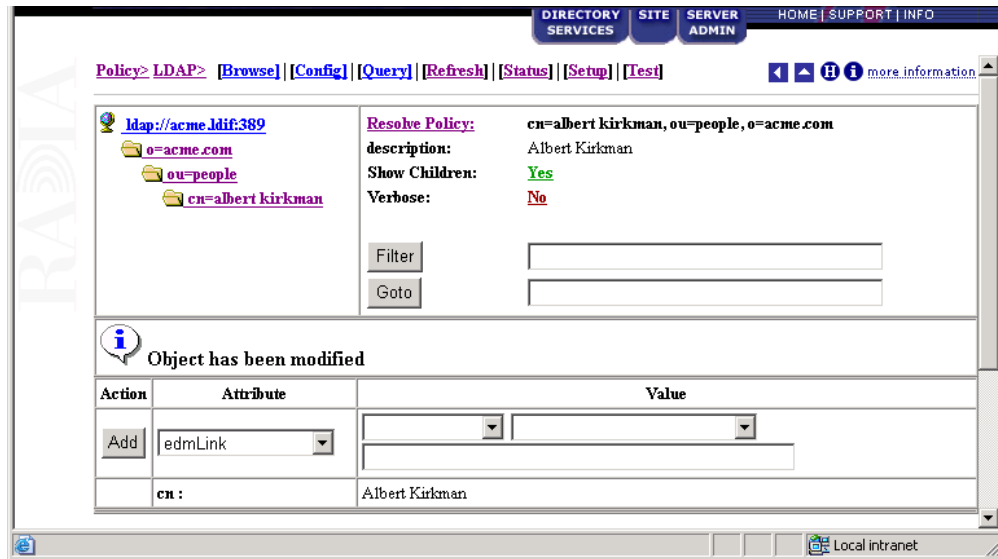
Figure 3.23 ~ cn=albert kirkman is selected for Policy management.

4. Make sure that **edmLink** is selected in the drop-down box for **Attribute**.

5. Type the complete distinguished name for the group or user that you want the selected user to connect to. In this example, we are connecting Albert Kirkman to ou=research,ou=development,o=acme.com.
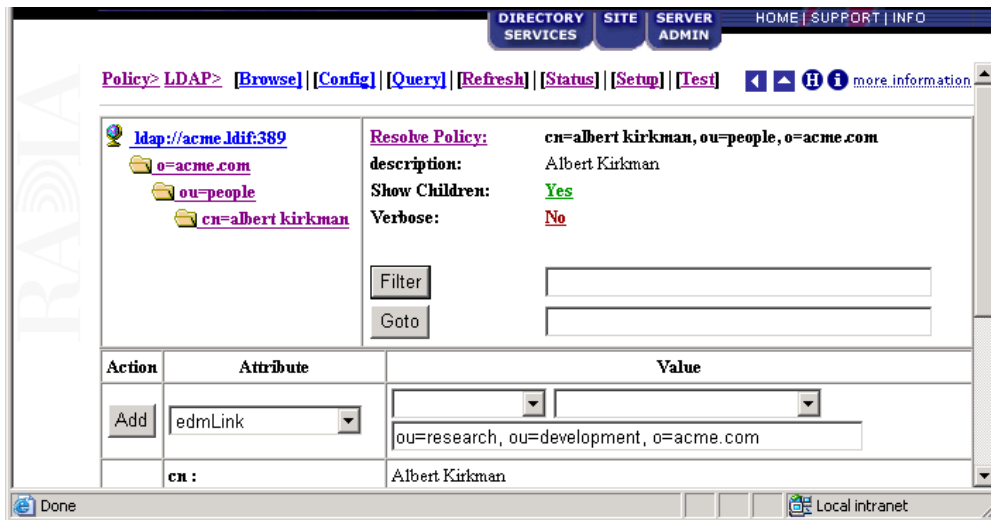
Figure 3.24 ~ Type the distinguished name of the object to connect to.

6. Click **Add** to add the link. The object has been modified and **edmlink** is added to the list of attributes for Albert Kirkman.
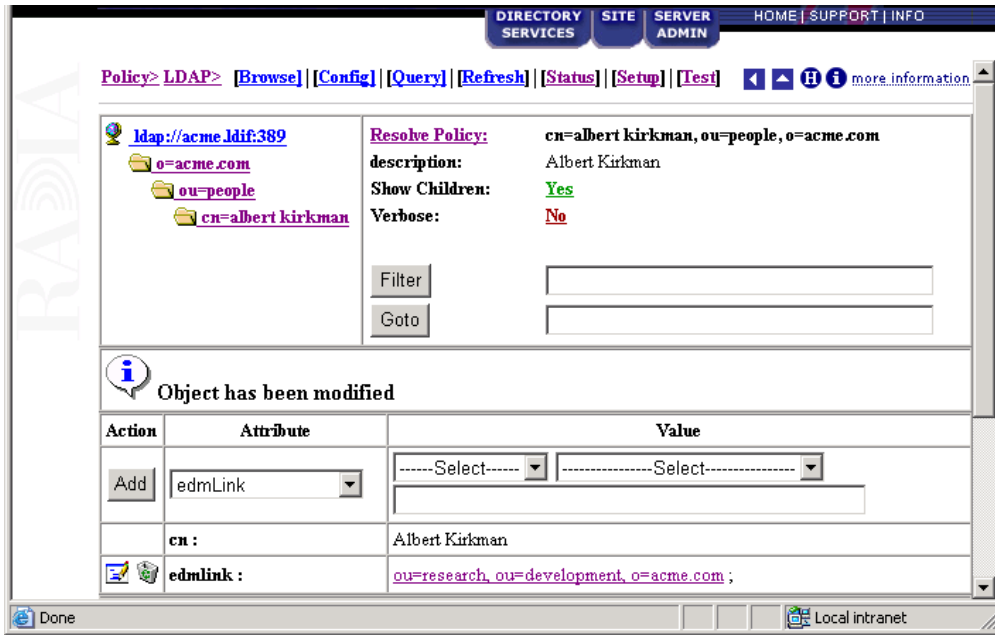
Figure 3.25 ~ The attribute is added to Albert Kirkman.

7. Verify that the object is inheriting the services from the object it is linked to by clicking **Resolve Policy**. In this example, Albert Kirkman should receive Stratus Pad from the research organizational unit.

Figure 3.26 ~ Resolve policy for the current object.

The user has been successfully linked to an additional organizational unit.

Policy Scope

By default, a subscriber inherits the policy from the parent of any groups it is linked to. This link can be through either the subscriber's directory service membership or through the use of the `edmLink` attribute. *Figure 3.27 ~ Acme Organization Directory structure* below shows a part of the Acme organization. It has three organizational units, Computers, Development, and People. Computers holds the Laptop container. Development includes the Product, QA, Research, and Support organizational units. People includes the actual users of the enterprise.
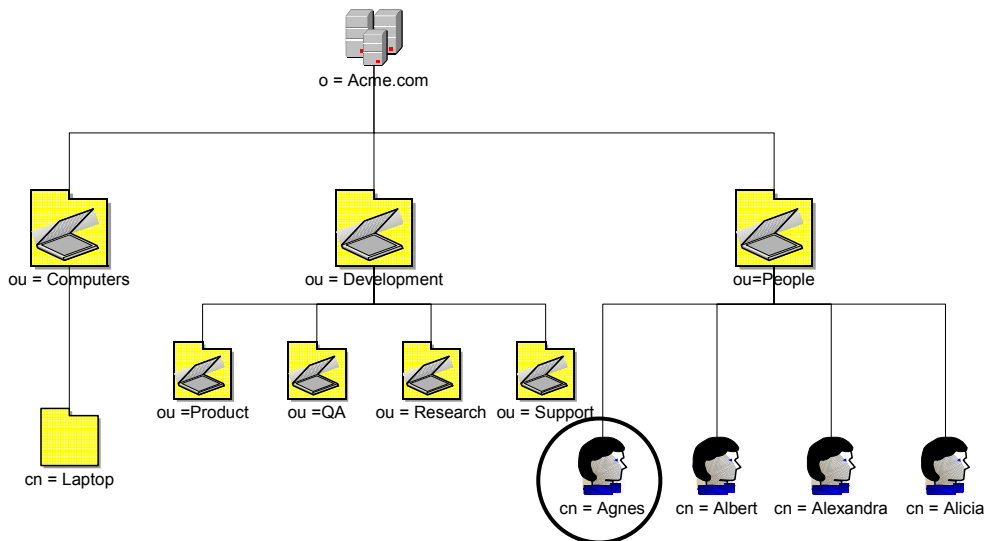
Figure 3.27 ~ Acme Organization Directory structure.

In *Figure 3.27 ~ Acme Organization Directory structure* above, Agnes will inherit the policy of the People organizational unit and the Acme organization.
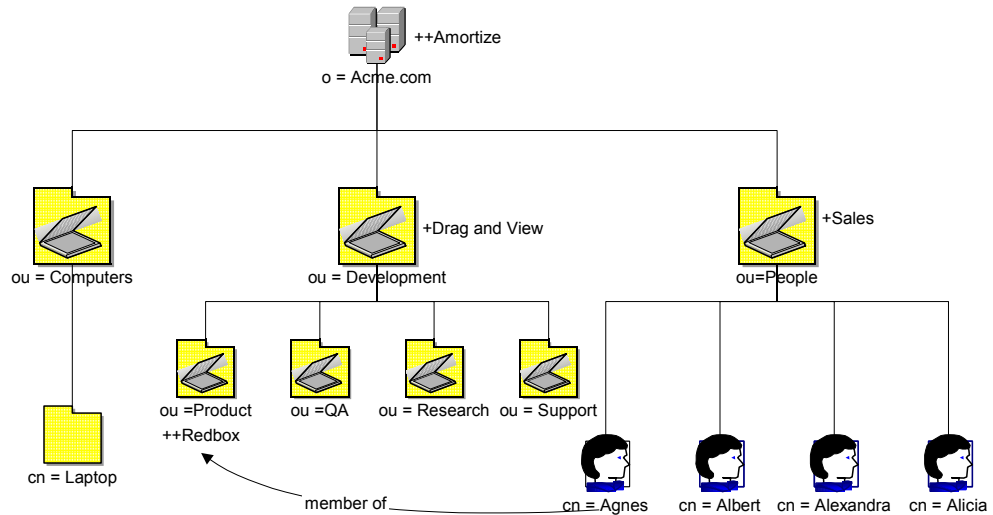
Figure 3.28 ~ Agnes is a member of Product organizational unit.

If Agnes is a member of the Product organizational unit, she will also inherit the policy from that unit and the Development organizational unit. In *Figure 3.28 ~ Agnes is a member of Product* above, Agnes would get Sales and Amortize because she is a part of the People organizational unit. Because Agnes is a member of the Product organizational unit, she would *also* inherit Redbox *and* Drag and View.

Suppose that you need Agnes to receive the services associated with the laptop container, but she is not linked to that container through directory services. Use edmLink to connect her to that container. See the procedure *To add a link* on page 70 for more information.
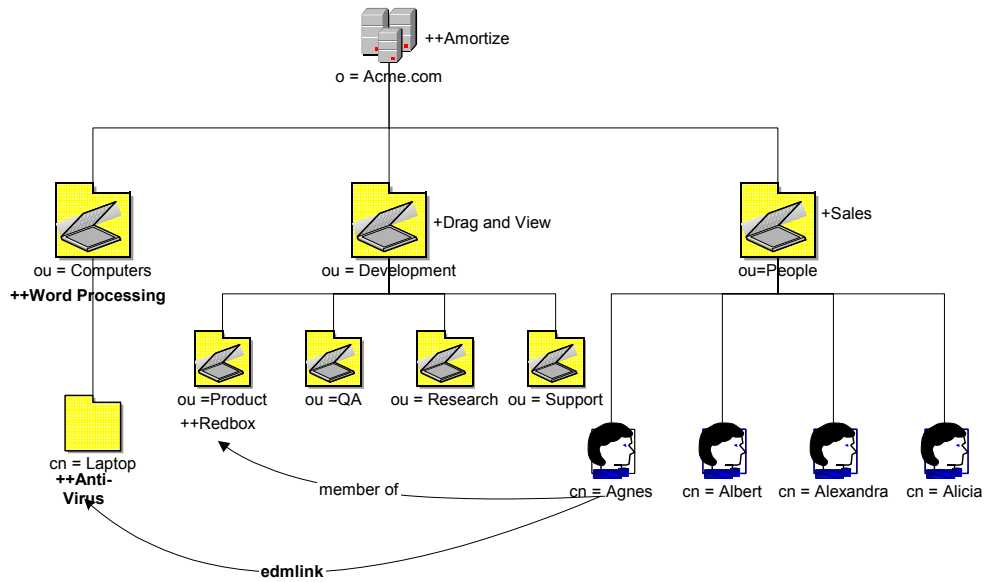
Figure 3.29 ~ Agnes is linked to the laptop container.

In *Figure 3.29 ~ Agnes is linked to the laptop container* above, Agnes will receive Anti-Virus because she has been linked to the laptop container. Since laptop is part of the Computers organizational unit, she will also get Word Processing. Now, she has a total of six applications.

Managing Policy Scope

If you do *not* want to inherit the policy from the parent objects, you can limit the Radia Policy Server's scope of resolution. You can do this either globally for the entire directory structure or for only specific objects. Manage the scope globally by modifying the Radia Policy Server's configuration file. Control policy scope for one object by using the `edmFlags` attribute.

Caution
Be sure that you have a thorough understanding of your directory structure. When designing a change to the scope of policy resolution, anticipate the result of your modifications *before* making the modifications.

Controlling Policy Scope Globally

The VIEW option allows you to control whether or not to continue up the directory tree to assign policy. Modify the VIEW option in the Radia Policy Server Configuration file, `pm.cfg`, to control the scope.

The syntax for the VIEW option is:

```
VIEW {
    <attr> {view}
    }
```

Where *attr* is one of the attributes listed in the LINKS configuration option in `pm.cfg`, and *view* is a list of LINKS the Radia Policy Server is allowed to see. An empty list means that there is no view when visiting an object from the specified attribute. This would result in following that link and not continuing. You can list as many or as few attributes as needed.

The default values for the LINKS configuration option are: `edmLink`, `memberof`, `groupmembership` and `aliasedobjectname`. When you look at a particular object such as a group or user through the Radia Policy Server interface, you will see only these attributes for that object. If you do not want Radia Policy Server to inherit the policy for any parents of an `edmLink` attribute, modify the VIEW option in `pm.cfg` like this:

```
VIEW {
    edmLink {  }
    }
```

This configuration with the empty brackets tells Radia Policy Server to follow `edmLink`, but not to inherit from any parents or any links contained within the object from that branch of the directory tree.

Looking back at the Acme organization example, suppose you want Agnes to receive policy for the laptop container, but not inherit any policy from the Computers organizational unit. In *Figure 3.29 ~ Agnes is linked to the laptop container* on page 77, Agnes will receive Anti-Virus because she has been linked to the laptop container, but she will *not* inherit Word Processing.

Similarly, if we wanted to follow a memberof attribute, and then not inherit from the parent objects, we would replace *edmLink* with *memberof*. The VIEW option would look like this:

```
VIEW {
    memberof {   }
    }
```

This configuration with the empty brackets tells Radia Policy Server to follow memberof, but not to inherit from any parents from that branch of the directory tree or any links contained within the object.

Finally, suppose that we only want to follow memberof relationships. The VIEW option would look like this:

```
VIEW {
    memberof {memberof}
    }
```

This configuration with the memberof in quotes tells Radia Policy Server to follow memberof, but not to inherit from any parents from that branch of the directory tree. When we follow a memberof relationship, we will continue to follow memberof relationships until we reach an object that does not contain a memberof relationship. In the figure below, Agnes will get Sales, Amortize. Then she will get Redbox because she is a member of Product. Since Product is a memberof laptop, she will get Anti-Virus. If Laptop had any memberof relationships, she would follow those relationships, too. Agnes will not follow any relationships other than memberof.
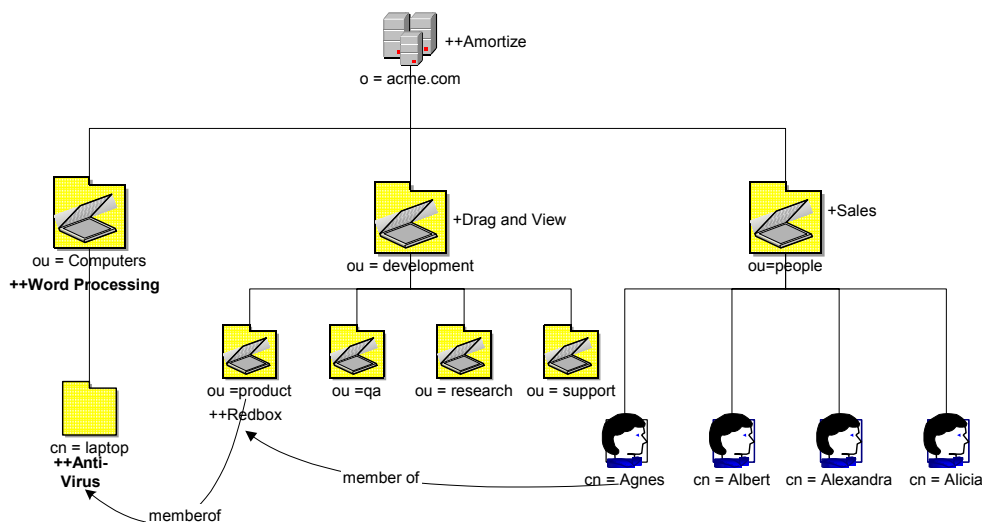
Figure 3.30 ~ Product is a member of the laptop container.

Controlling Policy Scope Locally (edmFlags)

You may want to limit the scope of policy resolution for only specific objects. To do this, use the edmFlags attribute. Edmflags is an optional, single-value integer attribute, and can contain any of the values shown in *Table 3.5 ~ edmFlags Values* on page 81 with a logical OR. Use these flags very sparingly, as they can have a profound impact on the clarity and function of the policy model.

In the example below, we will prevent Albert Kirkman from traversing up the directory tree to resolve his policy.

To use edmFlags

1. Open your Internet browser to either the Radia Management Portal page or the Radia Policy Server page.
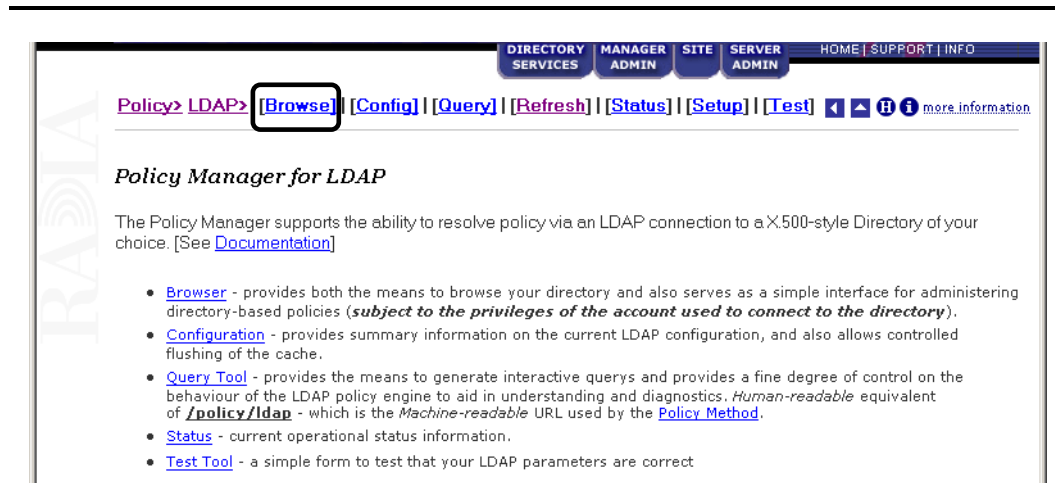
Figure 3.31 ~ Click Browse on the Radia Policy Server page.

2. Click **Browse**.

3. Click on an organizational unit in the bottom section of the policy management screen to select it or continue to select until you reach a particular user. In *Figure 3.32 ~ cn=albert kirkman is selected for Policy management* on page 81 we selected **albert kirkman** from the people organizational unit in acme.com.
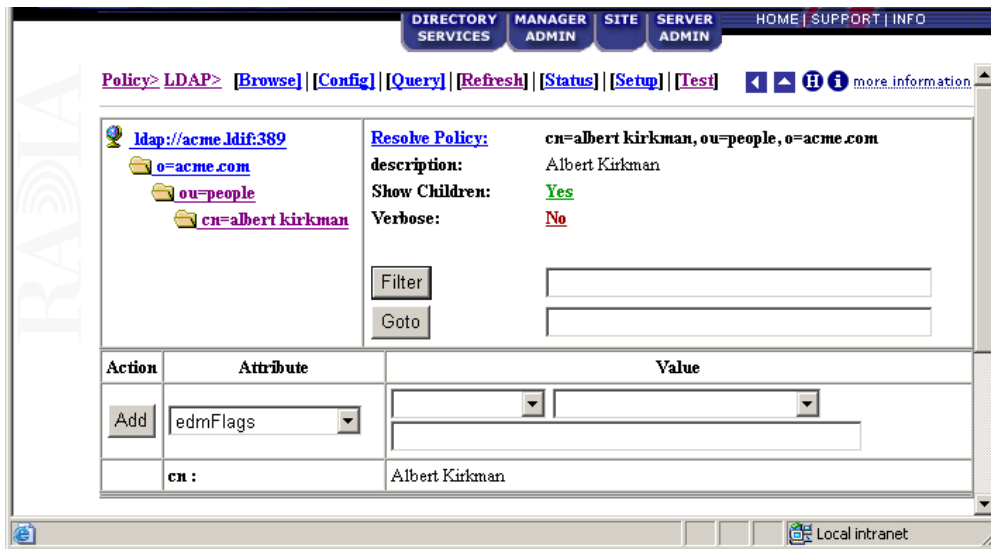
Figure 3.32 ~ cn=albert kirkman is selected for Policy management.

4. Make sure that **edmFlags** is selected in the drop-down box for Attribute.

5. Using *Table 3.5 ~ edmFlags Values* below, type the number that corresponds to the control that you want to exert over Albert Kirkman's policy.

Table 3.5 ~ edmFlags Values	
Value (Value Name)	**Description**
1 (NVD_SECEDE)	Use NVD_SECEDE to instruct the Radia Policy Server *not* to include any parent objects traversing up the object's branch of the directory tree. Its primary use is to support semi-autonomous units within an organization. For example, if your organizational unit is a member of a larger organization, and you do not want to resolve your policy above your organizational unit, use NVD_SECEDE. Set edmFlags to **1** for your organization unit.
2 (NVD_CONTINUE)	Use NVD_CONTINUE to ignore all of this object's attributes, and continue up the directory tree instead. The parent object is still processed, unless NVD_SECEDE is set. Set edmFlags to **2**.
4 (NVD_BREAK)	Use NVD_BREAK to abort the policy resolution and return the condition to the client. In this situation, the client device *should not* apply policy. It can be used to implement "change control freezes", to prevent policy changes being applied to certain parts of an organization. Set edmFlags to **4**.

Table 3.5 ~ edmFlags Values	
Value **(Value Name)**	**Description**
8 `(NVD_STRICT)`	Use NVD_STRICT to ignore memberOf attributes, and only process `edmFlags`, `edmPolicy`, and `edmLink`. Set `edmFlags` to **8**.

6. In this example, we want to only resolve the policy for Albert Kirkman and none of his parent objects. Therefore, we will type 1 in the **Value** entry box.
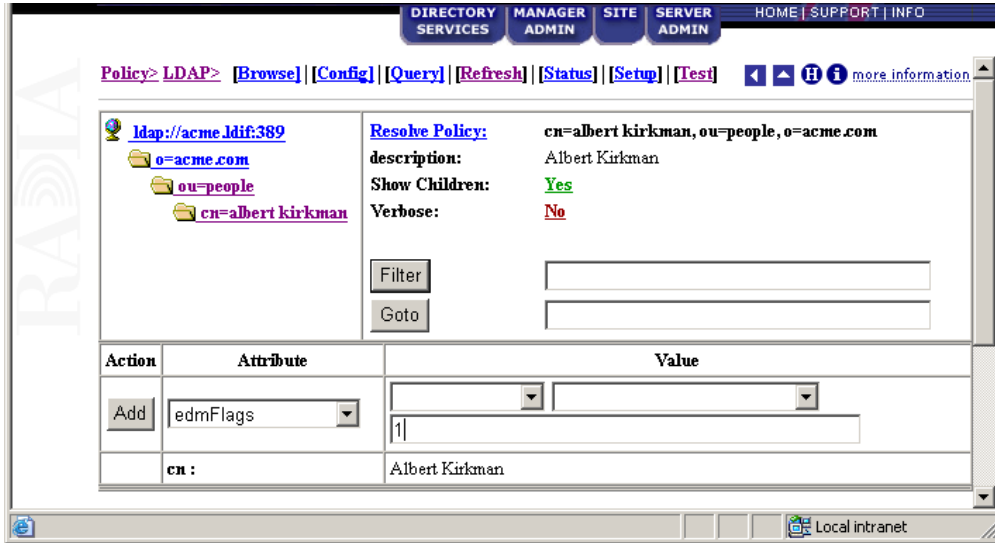
Figure 3.33 ~ Type the value for edmFlags.

7. Click **Add** to add the flag. The object has been modified and `edmFlags` is added to the list of attributes for Albert Kirkman.

Verify that the object is only resolving the policy for itself and none of its parents' policies by clicking **Resolve Policy**.

Log Files

To troubleshoot your Radia Policy Server, you may need to examine the log file. Within the logs directory for your Radia Integration Server, examine httpd-3466.log. This log is created when the Radia Integration Server starts up. It contains useful information if errors occur.

Summary

■   Configure the Radia Policy Server to connect to your directory services.

■   Configure your Radia Configuration Server to use the LDAP method.

■   Connect any users that you want to use Radia Policy Server for entitlements to the LDAP method.

■   Add your policy to the Radia Policy Server Web interface using your Internet browser.

■   You can control the scope of policy resolution globally using the VIEW option in the Radia Policy Server configuration file.

A

LDAP Discussion

This appendix provides more information on directory services for Radia Policy administrators needing additional information. It also includes descriptions of LDAP terminology, the use of substitution and expressions, and URLs used for Radia Policy Server.

LDAP Background

An LDAP directory is a hierarchically named tree of objects, where each object has a class (type) or classes, and contains potentially many named attributes, appropriate to its classes. Each attribute may contain multiple values.

It is outside the scope of this document to describe in any detail what an LDAP directory means. As a rapidly growing force in the systems management industry, many excellent sources exist for further background.

The Radia Policy Server is not concerned with such differences in interpretation—our only requirement is that the directory supports either the LDAP v2 or LDAP v3 protocols.

Radia Policy Server and LDAP

The LDAP Policy Extension, in conjunction with the Radia Policy Server, is intended to provide a scalable policy infrastructure, leveraging your existing investment in directories. The LDAP Policy Extension was developed to provide "Low Cost of Entry" to policy-based management, allowing you to start with a very simple policy model and incrementally grow the model as your

policies mature. The LDAP Policy Extension provides a clean integration with the standard repository for enterprise management information (LDAP), and allows an organization to leverage the information represented in its directories to deliver sophisticated policy-management to the many computing devices in its enterprise.

Radia Policy Server aimed at customers who have a detailed understanding of LDAP/X.500 directories, and an established directory infrastructure. The Radia Policy Server uses the LDAP protocol (version 2 or 3) (over TCP/IP) to speak to the customer's directory. This protocol encompasses all major directory products on the market, including the latest offerings from companies such as Novell, Microsoft, and Netscape.

The LDAP Policy Extension extends the Radia Policy Server with a number of features that enable you to represent your software management policy within your existing directory infrastructure and have this policy drive your Radia infrastructure to provide a comprehensive and sophisticated software management solution.

The extension makes policy resolution available via a URL utilizing the standard Radia Policy Server policy framework. It maintains a persistent LDAP connection to your corporate directory, and provides online HTML documentation and a number of interactive tools for discovering or diagnosing the policy outcome for target objects (typically users or machines) in your directory.

It is anticipated, but not required, that the Radia Policy Server hosting this extension be co-located on or near the directory to keep network latency to a minimum and enhance performance and manageability.

The LDAP Policy Extension understands the standard relationships that exist in a directory between different objects (parent-child, memberOf). In addition to these standard relationships, three additional attributes may be used:

- `<pfx>Flags`
  controls various subtle aspects of the policy resolution. See *Controlling Policy Scope Locally (edmFlags)* on page 80.
- `<pfx>Link`
  allows you to specify additional, potentially dynamic or conditional relationships. See *Controlling Policy Scope Locally (edmFlags)* on page 80.
- `<pfx>Policy`
  allows you to define resultant strings that will be netted out during policy resolution. See *Adding a Policy (EdmPolicy)* on page 61.

By default the prefix used is "edm", but alternatives may be used to allow your directory to support multiple concurrent policy frameworks for different purposes.

The LDAP Policy Extension starts at the specified DN, and walks the entire tree of relationships that the object has with other objects, accumulating policy attributes. Then it evaluates all conditional policies, and finally resolves any conflicting policies, using a straightforward should/may, grant/deny model.

Terminology

Before using directory-based policy management with Radia Policy Server, it is important to establish some terminology that is used throughout this discussion.

■ **Should**
   This is used to describe a mandatory or *required* policy.

■ **May**
   This is used to describe a desired or *advisory* policy.

■ **Policy**
   This is a string that is used to *represent* a *desired* outcome. The Radia Policy Server does not impose any particular interpretation upon this. When used in conjunction with the LDAP Adapter, the adapter will *interpret* this as the name of an application defined within Radia.

■ **Relationship (link)**
   Two directory objects are said to be *related* if one can be reached from the other, directly or indirectly. Examples of relationship include parent-child, and group membership (a user is *related* to the group he is a member of). Relationships are unidirectional.

■ **MemberOf**
   This is used to describe a *relationship* between two objects. Many common directories support an attribute called *memberOf* that embodies this relationship, typically between users and groups.

Substitution

Two forms of substitution are provided:

■ Current Object Attributes: `<<nameOfAttr>`, or

■ Inbound Object Attributes: `<<in.nameOfAttr>>`

The former allows you to construct expressions based upon the value of another attribute in the current object (same one that contains the edmLink or edmPolicy), for example,

```
edmLink: cn=<<homePC>>, cn=Computes, o=Acme.
edmLink: cn=wnt001, cn=Computers, o=Acme ; <<homePC>> == "wnt001".
```

The latter allows you to reference attributes that were supplied as input to the policy resolution, for example:

```
edmPolicy: +Microsoft Office ; <<in.os>> == "NT"
```

Currently the minimum attributes that will exist are in.os (operating system), in.uid (User ID), and in.host (Host Computer).

Expressions

The expressions are implemented as Tcl (`www.scriptics.com`) expressions, where instead of using $myVar you would use `<<myAttribute>>`. A simplified summary of valid expressions is provided in Table A.1 on page 88. Most of the standard C language expression operators are valid.

Table A.1 ~ Expressions

Expression	Meaning
A && B	Logical AND
A || B	Logical OR
!A	Logical NOT
<<myAttr>> == "Hello"	Test for equality (case-sensitive)
<<myAttr>> != "Hello"	Test for inequality
<<myAttr>> < 55	Numerical comparison for less than
<<myAttr>> >= "Hello"	Dictionary comparison for greater than or equal to (C locale)

There are also a small number of specialized functions.

Table A.2 ~ Specialized Function Examples

Example	Meaning
[memberOf "ou=Accounting, o=Acme"]	Yields TRUE if the DN specified is part of your policy model.
[parent <<dn>>] == <<aSpecialDN>>	Yields TRUE if the parent DN of the current object is the same as the "aSpecialDN".

The LDAP Extension URL Namespace

The LDAP extension provides the following special purpose URLs:

Table A.3 ~ LDAP Extension URL Namespace	
URL	**Description**

/policy/ldap?<x-url encoded query>	Perform machine-readable policy resolution. The query arguments should be an attribute value list of inbound attributes, formatted in accordance with the X-URL encoding specification. The following attributes are currently supported and interpreted by the LDAP Policy Extension:
	
	• **dn**
	   the distinguished name or LDAP URL to perform policy resolution upon. (REQUIRED)
	• **phase**
	   the value may be specified as "1", "2", or "3", to view the intermediate stages of policy resolution. (default=3)
	• **prefix**
	   the value is the prefix to use when searching the directory for policy related attributes, i.e., `<pfx>Policy` or `<pfx>Link`. (default=edm)
	• **debug**
	   the value is the log level to use for this single query, a value of 9 or above will generate detailed logging in the Radia Policy Server log file. (no default)

/status/ldap	Return an overview of the current status of the extension.
/status/ldap/all	Return all available status information on extension.
/status/ldap/cache	Return information on cache.
/status/ldap/stats	Return statistics on usage of extension.
/admin/ldap/flush?dn=<dn>	Force a flush of the cache. If no dn, or an empty dn, is supplied, then the entire cache is flushed. Otherwise, just the specified dn is flushed.
/admin/ldap/reset	Reset connection to directory (forces a flush and reconnect).
/ldap/config.tsp	Summary configuration page, and interactive controls for resetting cache and connection.
/ldap/browse.tsp?dn=<dn>	Directory Browser and Policy Editor.
/ldap/query.tsp?dn=<dn>	Interactive Policy Resolver—simple diagnostic page allowing you to interactively submit policy requests and see the policy outcome, as well as the steps that led to that outcome, in a friendly formatted HTML page.
/ldap/test.tsp?dn=<dn>	This URL can be used to test connections to arbitrary directory servers, and is useful when diagnosing problems with authentication and directory access.

B

Use Existing LDAP Attributes

The goal of this feature is to allow Radia customers to implement the Radia Policy Server without requiring schema changes. This can be accomplished by using an existing directory service attribute to embed the required Radia attributes and their values.

Cautions
Do not implement this feature with a directory that already has the necessary Radia attributes. The feature will *not* function properly. Read *Adding Radia Policy Attributes* on page 42 before using an existing LDAP attribute.
This feature should *only* be used when it is *not* possible to make the necessary LDAP schema changes as shown in *Adding Radia Policy Attributes on page 42*

To use this feature, you must have an unused multi-valued LDAP attribute that already exists in the directory schema that can exist in *any* object that will have policy assignments. Use the EMBED configuration option in the Radia Policy Server's configuration file, `pm.cfg`. The value of EMBED must be the name of an attribute that already exists in the schema of your LDAP directory. The attribute should be one that is allowed to exist in all objects for which policy will be assigned.

The attribute should be multi-valued and of type string. The embedded data will be stored in multiple values of the attribute – one embedded policy per value. The original contents will be maintained along with any policies assigned to the object.

Suppose you are going to use an already existing attribute called "displayname". Add the following line to `pm.cfg`:

```
EMBED              {displayname}
```

By default, the EMBED options assumes that the displayname attribute is multi-valued.

C

Domain Filtering

If you are using Radia Policy Server to create entitlements in your enterprise, you can filter out which domains the Radia Policy Server will assign services from based on connect parameters.

If you are using Radia Policy Server with Radia Patch Manager, you will want to separate resolution of regular software services from those for Radia Patch Manager. Radia Policy Server filters services based on the dname passed on the radskman command line. The Radia Policy Server configuration file, pm.cfg, contains filter settings in format:

`DNAME=<DOMAIN NAME>   { rule }`

Where the DOMAIN NAME is the value passed in dname by RADISH.   In the case of a Radia Patch Manager client, this will be the dname parameter of radskman. Dname should be "patch". If the filter name passed in dname is not found in pm.cfg, then the filter `DNAME=*` will be used. The minimum version requirement for Radia Policy Server is version 3.2.1.

The default configuration for the for these filters is shown in the figure below:

```
DNAME=*           { * !PATCHMGR !OS }
DNAME=PATCH       { PATCHMGR }
DNAME=OS          { OS }
```

Figure C.1 ~ Configure Radia Policy Server filters.

In this configuration the default rule (*) will ignore PATCHMGR and OS domains and allow everything else as denoted by the use of "!". PATCH and OS rules allow only policies for PATCH and OS domains respectively.   If for instance, we wanted to allow any policies for OS manager resolution we would change the last filter to: `DNAME=OS   { * }`.

Lists

Figures

Tables

Procedures

Index