

Oracle® COREid
Access and Identity

Administration Guide

*Volume 1: COREid and Common
Administration*

**10g Release 2 (10.1.2)
Part No. B19008-01**

May 2005

ORACLE®

Copyright © 1996-2005, Oracle. All rights reserved. US Patent Numbers 6,539,379; 6,675,261; 6,782,379; 6,816,871.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Oracle COREid Access and Identity products includes RSA BSAFE™ cryptographic or security protocol software from RSA Security. Copyright © 2003 RSA Security Inc. All rights reserved. RSA and RC4 are trademarks of RSA Data Security. Portions of Oracle Internet Directory have been licensed by Oracle Corporation from RSA Data Security. This product includes software developed by the Apache Software Foundation (<<http://www.apache.org/>>). Copyright © 1999-2003 The Apache Software Foundation. All rights reserved. Copyright © 2003 The Apache Software Foundation.

This program contains third-party code from Apache. Under the terms of the Apache Software License, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Apache software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the Apache software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Apache.

* The Apache Software License, Version 1.1

- *
 - * Copyright (c) 2000 The Apache Software Foundation. All rights reserved.
 - *
 - * Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 - * 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 - * 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 - * 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:
 - * "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."
 - * Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

* 4. The names "Apache" and "Apache Software Foundation" must
* not be used to endorse or promote products derived from this
* software without prior written permission. For written
* permission, please contact apache@apache.org.

*
* 5. Products derived from this software may not be called "Apache",
* nor may "Apache" appear in their name, without prior written
* permission of the Apache Software Foundation.

*
* THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.

* =====

*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.

*
* Portions of this software are based upon public domain software
* originally written at the National Center for Supercomputing Applications,
* University of Illinois, Urbana-Champaign.

*/

Contents

Preface	17
Intended Audience	17
COREid Documentation.....	18
Typographical Conventions	19
Contact Information.....	19
Corporate Headquarters	19
Before Contacting Customer Care	19
Accessing the Customer Care Knowledge Base	20
Section I: Introducing NetPoint Administration	21
Chapter 1 Preparing for NetPoint Administration	23
Prerequisites	23
About COREid System Configuration and Administration	24
COREid System Components	24
Review of COREid Installation and Setup	25
About Configuring the COREid System	26
About Managing the COREid System	29
About Access System Configuration and Administration	29
Access System Components	30
Review of Access System Installation and Setup	31
About Configuring the Access System	31
About Managing the Access System	33
Introduction to Using NetPoint	33
About Login	33
Logging In to the COREid System	34
Logging In to the Access System	36
Screen Functional Areas	38
Online Help	38
The About Link	39
The Selector	40
Logging Out	41

Chapter 2	Specifying COREid System Administrators	43
	About COREid System Administrators	43
	Specifying Administrators	46
	Deleting Administrators	47
	Delegating Administration.....	47
	About Delegating Administration	48
	Delegated Administration Models	49
	Adding Delegated Administrators	52
	Adding Substitute Administrators	55
	 Section II: Configuring the COREid System.....	 57
Chapter 3	Making Schema Data Available to NetPoint.....	59
	About Object Classes	60
	About Template Objects and Provisioning	61
	The Process for Configuring Schema Data	61
	Objects Configured During Installation	62
	Structural and Auxiliary Object Classes in NetPoint	62
	Template Object Classes	64
	Object Class Types	64
	Viewing Object Classes.....	65
	Modifying Object Classes.....	67
	Selecting a Class Attribute	67
	Changing the Structural Object Class	68
	Adding Object Classes	69
	How Auxiliary Classes Are Used	70
	Deleting Object Classes	71
	About Object Class Attributes.....	71
	About Configuring Attributes	72
	Attribute Data Types	73
	Attribute Semantic Types	74
	Attribute Display Types	78
	Viewing Attributes.....	80
	Configuring Attributes.....	81
	Using Rules and Lists	83
	Localizing Attribute Display Names	86
	Search Filters for the Object Selector Display Type	87

Creating a Search Filter for the Object Selector Display Type	88
Search Filters for Multiple Target Object Classes	89
Deleting a Search Filter	89
Usage of Rules and Filters	89
Configuring Other Display Types	93
Configuring Derived Attributes	94
Attributes Configured on a Per-Application Basis	97
Chapter 4	
 Configuring User, Group, and Organization Manager.....	99
About User, Group, and Organization Manager	100
Configuring Tabs.....	101
Viewing and Modifying Tab Configuration Information	101
Localizing Tabs	104
Adding a Tab to the Organization Manager	105
Specifying the Search Attributes on a Tab	106
Viewing, Modifying, and Localizing Search Result Attributes	106
Adding Auxiliary and Template Object Classes to a Tab	108
Adding Special-Purpose Object Classes to a Group Tab	109
Configuring Group Manager Tab Options	110
Deleting a Tab in Organization Manager	112
Ordering the Tabs in Organization Manager	112
Configuring Tab Profile Pages and Panels	113
Use of LDAP and Template Objects on a Panel	113
Configuring the Header Panel	114
Viewing Panels	115
Adding, Modifying, Localizing, and Deleting a Panel	116
Ordering the Panels	120
Viewing Group Type Panels	121
Adding, Modifying, Localizing, and Deleting a Group Type Panel	122
Modifying and Localizing Attributes Displayed on a Panel	124
Allowing Users to View and Change LDAP Data	126
About the Searchbase	126
Guidelines for Setting the Searchbase	127
Setting the Searchbase	128
Setting Up Disjoint Searchbases	131
Writing LDAP Filters Using Query Builder	132
Building Advanced LDAP Filters Using QueryBuilder	135
About View and Modify Permissions	137
Setting and Modifying LDAP Attribute Permissions	138
Keys for Selecting Multiple Attributes	141

Evaluation of LDAP Attribute Permissions	142
Examples of Configuring an Application	143
Displaying Photos in User Profiles	143
Enabling the Location Tab in Organization Manager	146
The Right to Create Groups in Group Manager	146
End-User Scenarios	147
Managing Group Members in Group Manager	147
Searching for Group Members	147
Customizing Search Results for Group Members	149
Deleting Group Members	150
Adding Group Members	150
Managing Group Subscriptions	151
Subscribing to Groups	152
Configuring Logging and Auditing Policies	153
Viewing Logging and Auditing Policies	153
Modifying Logging and Auditing Policies	154
Generating Reports	156
Configuring Reports	156
Viewing, Modifying, Localizing, and Deleting Reports	159
Advanced Configuration	161
Expanding Dynamic Groups	161
Modifying the Default Searchbase Scope	162
Simplified Attribute Permissions for a Group	163
Setting Container Limits in Organization Manager	165
Chapter 5	
Chaining COREid Functions Into Workflows.....	171
About Workflows.....	171
Typical Workflow Examples	172
Advanced Workflow Options	173
Workflow Types	173
Creating Workflows	174
How Users Access Workflows in a COREid Application	175
LDAP Versus Template Attributes in a Workflow	178
Workflow Types, Steps, and Actions	179
About Workflow Steps	180
About Step Actions	182
Descriptions of Step Actions	186
About Subflows	189

Using the QuickStart Tool	190
Creating a Self-Registration Workflow Using the Quickstart Tool	193
Using the Workflow Applet	194
Starting a New Workflow Definition	196
Defining an LDAP Target for Create Object Workflows	199
Defining the First Step in a Workflow	201
Defining Step Attributes	203
Defining Subsequent Steps	206
Committing Workflow Steps	208
Enabling the Workflow	208
Testing the Workflow	209
Example of Defining a Workflow	209
Defining a Subflow	210
Associating a Subflow with a Workflow	211
Approving Subflow Steps	212
Advanced Workflow Ticket Routing	212
Configuring Workflow Actions for Advanced Ticket Routing	212
About Notifying Newly Assigned Step Participants	213
Specifying Dynamic Participants	214
Specifying Surrogates	221
Enabling Time-based Escalation	224
Performing Asynchronous Operations	228
Notes on Asynchronous Workflows	228
Using a Workflow	229
Invoking a Workflow	229
Finding and Processing a Ticket	230
Deactivating and Reactivating Users	232
Reactivating a Deactivated User	232
Monitoring a Workflow	233
Archiving Requests	234
Deleting Requests	234
Preventing Other Administrators from Working on a Workflow Ticket	235
Managing Workflows.....	235
Viewing and Exporting a Workflow Summary	235
Copying a Workflow	237
Modifying a Workflow	237
Deleting a Workflow	238
Exporting Workflows	238
Viewing Workflow Panel Settings	239
Modifying the Appearance of Workflow Panels	241

	Localizing Workflow Panels	242
	Workflow Performance	243
	The NetPoint Identity Administrator's Modify Rights	244
	Advanced Workflow Options	244
	Pre and Post Actions	244
	External Actions	245
	Customization of Data and Actions in a Workflow	245
	Adding Roles to a Workflow	246
	Creating a Self-Registration Workflow.....	247
	Creating a Location Workflow.....	250
Chapter 6	Provisioning External Applications from COREid.....	253
	About Provisioning Application Accounts	254
	Summary of Provisioning Using a Workflow.....	254
	About Template Objects and Provisioning	256
	About Template Object Data and Workflows.....	256
	Object Template Configuration.....	257
	Format of the Object Template File	257
	How Template Objects Appear in the COREid System	259
	Elements in an Object Template File	260
	Sample Object Template File	262
	Creating an Identity Event Plug-In for Template Attributes	264
Chapter 7	Configuring and Managing the COREid System	265
	Configuring Styles for COREid Applications.....	266
	Viewing a Style	267
	Adding a Custom Style Directory	267
	Deploying a Style	270
	Changing a Style Name	271
	Modifying a Style	271
	Deleting a Style	271
	Setting the Default Style	272
	Configuring Multiple Languages for NetPoint.....	272
	Selecting a Language to Display	274
	Language Evaluation Order for NetPoint Applications	275
	Configuring COREid Server Settings	276
	Configuring Session Timeout	278

Configuring Licenses	279
Updating License Keys on Multiple COREid Servers	280
Customizing Email Destinations	280
Configuring a Mail Server	281
Managing Caches	283
Managing Multiple Languages	283
Managing COREid Servers.....	284
Setting Up Multiple COREid Servers	284
Adding a COREid Server	285
Viewing and Modifying COREid Server Parameters	288
Deleting COREid Server Parameters	288
Managing a COREid Server Service from the Command Line	289
Managing Directory Server Profiles	290
About LDAP Directory Server Profiles	290
Creating an LDAP Directory Server Profile	291
Viewing an LDAP Directory Server Profile	298
Modifying an LDAP Directory Server Profile	298
Rerunning NetPoint Setup Manually	299
Adding Database Instances to LDAP Directory Server Profiles	301
Deleting an LDAP Directory Server Instance	305
Managing RDBMS Profiles	305
Adding or Modifying an RDBMS Profile	306
Adding or Modifying an RDBMS Database Instance	309
Configuring WebPass	310
Viewing a Configured WebPass	311
Adding or Modifying a WebPass	312
Modifying WebPass Details	314
Removing a WebPass	315
Modifying a WebPass from a Command Line	315
Managing Associations between COREid Servers and WebPass	318
Disassociating a WebPass from a COREid Server	319
Configuring Password Policies	320
Order of Password Policy Evaluation	321
Managing Password Policies	321
Configuring Lost-Password Management for the COREid System	327
Implementing Password Policies in the Access System	328
Configuring the Access Server SDK for the	

	COREid System	333
Section III: Performing Common Administrative Tasks		335
Chapter 8	Changing Transport Security Modes.....	337
	About Transport Security Modes	337
	Transport Security Mode Between NetPoint Components	339
	About CA Certificates	341
	Changing Transport Security for the COREid System	342
	Transport Security Mode Changes for the COREid System	344
	Changing to Simple Transport Security Mode	345
	Changing to Cert Transport Security Mode	346
	Changing Transport Security Modes for the Access System	349
	Transport Security Mode Changes for the Access System	350
	Changing to Open Transport Security Mode	353
	Changing to Simple Transport Security Mode	354
	Changing to Cert Transport Security Mode	356
	Transport Security Changes for Directory Servers.....	361
	Changing Transport Security Passwords	362
	Importing Multiple CA Certificates	365
	Changing Access Server Security Password	366
	Cloned and Synchronized Components	366
Chapter 9	NetPoint Reporting.....	367
	About NetPoint Reporting.....	367
	Report Types	368
	Data Sources	369
	Data Output	370
	Output Configuration	370
	Data Uses	370
	Summary of NetPoint Reporting Features.....	371
Chapter 10	Logging	373
	About Logging and Log Levels	373
	Log Levels	374
	About Log Configuration Files	375
	Log Configuration File Paths	376

Log Configuration File Names	376
Modifying a Log Configuration File	377
About Log Writers	380
Log Configuration File Structure	381
About XML Element Order	383
Controlling Logging Levels.....	384
About Log Handler Precedence	385
Log Configuration Parameters	386
Default Log Settings	388
Configuring Logs in the COREid System Console.....	390

Chapter 11

Auditing	395
About NetPoint Auditing.....	395
Audit Output Considerations	396
Audit Security Considerations	396
Audit Performance Considerations	396
Static Audit Reports	397
Dynamic Audit Reports	398
Controlling Audit Output.....	399
About NetPoint Audit Options	400
NetPoint Auditing Requirements.....	403
Audit-to-Database Requirements	403
Audit-to-Database Architecture.....	405
About ODBC Data Source Definitions	406
About ODBC Drivers	407
About RDBMS Profiles	408
About the NetPoint Audit Database	409
About the Oblix/Crystal Repository	410
Setting Up File-Based Auditing	412
Setting Up Database Auditing.....	416
Setting Up Your NetPoint System for Database Auditing	417
Setting up the Audit Database	417
Configuring NetPoint Auditing	437

	Setting up NetPoint Audit Reports.....	451
Chapter 12	SNMP Monitoring.....	455
	Prerequisites.....	456
	About NetPoint SNMP and Agents.....	456
	The NetPoint SNMP Agent	457
	About the NetPoint MIB and Objects.....	457
	MIB Index Fields	459
	COREid Server MIB Objects	459
	Access Server MIB Objects	465
	Enabling and Disabling SNMP Monitoring.....	471
	Setting Up SNMP Agent and Trap Destinations.....	472
	Changing SNMP Configuration Settings	474
	Logging for SNMP	476
	NetPoint SNMP Messages	476
	Discrepancies Between Netstat and SNMP Values	482
	Configuring the Shutdown Interval	483
	Section IV: Appendices and Index	485
Appendix A	Deploying NetPoint with Active Directory.....	487
	Setting Up Directory Profiles and Searchbases	488
	Defining Directory Server Profiles for Remaining Domains	488
	Setting Up Disjoint Searchbases	489
	Configuring Group-Search Read Operations (Optional)	490
	Authentication and Authorization with Active Directory	491
	Parent-Child Authentication	491
	Parent-Child Authorization	492
	ObMyGroups Action Attribute	492
	Configuring the credential_mapping Plug-In	493
	Configuring SSO for Use with Active Directory	495
	About Search Filters	497
	Configuring NetPoint for .NET Features.....	497
	Troubleshooting.....	498
	Active Directory Search Halts	498

	Adding Members to Static Groups Causes the Group Size to Shrink	498
	Microsoft Resources	498
Appendix B	Configuring NetPoint for ADSI	499
	About ADSI with NetPoint	499
	Recommendation	500
	COREid System ADSI Configurations	501
	Pure ADSI with ADSI Authentication	501
	Mixed ADSI with LDAP Authentication	502
	NetPoint COREid ADSI Configuration Files	503
	Access System ADSI Configurations.....	506
	Pure ADSI with ADSI Authentication	507
	Access System ADSI Configuration Files	508
	Configuring ADSI for the COREid System.....	510
	Enabling ADSI for a Default Directory Profile	510
	Enabling ADSI for Other Directory Profiles	511
	Configuring ADSI for the Access System	513
	Enabling LDAP Authentication for the Access Server	514
	Changing the pageSize Parameter.....	515
	Troubleshooting	516
	ADSI Cannot Be Enabled for this DB Profile	516
Appendix C	Configuring NetPoint for Active Directory with LDAP	517
	Overview	517
	Setting Up the Access Manager for LDAP.....	518
	Setting Up the Access Server for LDAP	519
	Setting Active Directory Timeouts for LDAP	520
	Enabling LDAP Authentication with ADSI	521
Appendix D	Implementing .NET Features with NetPoint	523
	Resolving Ambiguous Names.....	523
	About ANR Attributes, Searches, and Results	524
	Configuring NetPoint for ANR	525
	Configuring NetPoint for Dynamically Linked Auxiliary Classes	529
	Adding Attributes Dynamically	530
	Adding Attributes for a Group	531

Enabling Fast Bind for NetPoint Authentication.....	534
Enabling Impersonation.....	536
Setting Up Integrated Windows Authentication	537
Enabling IWA on the WebGate Web Server	538
Configuring the WebGate for IWA	539
Creating an IWA Authentication Scheme in NetPoint	539
Testing IWA Implementation	540
Using Access System Password Management.....	540
Using Managed Code and Helper Classes	541
Integrating NetPoint with Authorization Manager Services	542
Integrating NetPoint with Passport Authentication	542
Integrating NetPoint with Smart Card Authentication	542
Integrating the NetPoint Security Connector for ASP.NET	543
Troubleshooting.....	543
Active Directory Search Halts	543
Microsoft Resources.....	544
Index	545

Preface

The Administration Guide is divided into two volumes. This book, *Volume 1*, provides information on configuring COREid to read and make use of data in your directory, configuring COREid applications to display directory data, assigning read and write permissions to users, defining workflows for activities such as creating users, defining access controls, using COREid to control user access to applications and data, and configuring single sign-on.

Note: Oracle *COREid* was previously known as *Oblix Netpoint*. All legacy references to *Oblix* and *NetPoint*, for example, in screen shots, illustrations, and documentation titles, should be understood to refer to Oracle and *COREid*, respectively.

This Preface covers the following topics:

- “Intended Audience” on page 17
- “COREid Documentation” on page 18
- “Typographical Conventions” on page 19
- “Contact Information” on page 19

Intended Audience

This guide is intended for the COREid Administrators assigned during installation and setup, as well as Master Identity Administrators and Delegated Identity Administrators. Administrators configure the rights and tasks available to other administrators and end users. A COREid Administrator, the highest level administrator, is selected during COREid System setup. The COREid Administrator delegates responsibilities to other administrators, as described in this book.

This document assumes that you are familiar with your LDAP directory and Web servers.

COREid Documentation

The manuals that are available for this release include:

Introduction to COREid—Provides an introduction to COREid, a road map to COREid manuals, and a COREid glossary of terms.

COREid Release Notes—Provides up-to-the minute details about the latest COREid release.

COREid Installation Guide—Explains how to install and configure the COREid components.

COREid Upgrade Guide—Explains how to upgrade earlier versions of COREid to the latest version of COREid.

COREid Administration Guide—Explains how to configure COREid applications to display information stored in the directory, how to assign view and modify permissions for data displayed on the COREid applications, and how to assign access controls to users.

COREid Deployment Guide—Provides information for people who plan and manage the environment in which COREid runs. This guide covers capacity planning, system tuning, failover, load balancing, caching, and migration planning.

COREid Customization Guide—Explains how to change the appearance of COREid applications and how to control COREid by making changes to operating systems, Web servers, directory servers, directory content, or by connecting CGI files or JavaScripts to COREid screens. This guide also describes the Access Server API and the Authorization and Authentication Plug-in APIs.

COREid Developer Guide—Explains how to create AccessGates and how to develop plug-ins. This guide also provides information to be aware of when creating CGI files or JavaScripts for COREid.

COREid Integration Guide—Explains how to set up COREid to run with third-party products such as BEA WebLogic, the Plumtree portal, and IBM Websphere.

COREid Schema Description—Provides details about the COREid schema.

Online Help is available from each COREid screen.

Typographical Conventions

COREid manuals use the following typographical conventions:

- When you are instructed to select elements sequentially, the actions are separated with angle brackets, as shown below:

Click System Admin > System Configuration > View Server Settings.

- Paths to a file are shown using syntax for either the Unix or Windows platform:

```
/COREid_install_dir/identity/oblix/logs/debugfile.lst
```

```
\COREid_install_dir\identity\oblix\logs\debugfile.lst
```

where *COREid_install_dir* refers to the directory where the component, in this case, the COREid Server, is installed.

Contact Information

For a list of contacts including corporate offices world wide, sales, and other details, visit the Oracle Web site at:

<http://www.oracle.com>

You can contact Oracle with questions or comments as follows:

Customer Care—<http://www.oracle.com/support/contact.html>

Corporate Headquarters

Oracle maintains offices world wide. Oracle corporate headquarters is located at:

500 Oracle Parkway
Redwood Shores, CA 94065
Phone: (650) 506-7000

Before Contacting Customer Care

Before contacting Customer Care, please have available the following:

- Oracle product name and version number
- Type of computer and operating system you are using

Accessing the Customer Care Knowledge Base

For more information about using COREid, see the Oracle Customer Care Knowledge Base. To access the Knowledge Base, you need a login name and password, which you can obtain from your Oracle sales representative.

To access the Knowledge Base:

1. Enter the following URL in your browser and press Return.
`http://www.oracle.com/support/contact.html`
2. Click the phrase, Login to the Oracle PremiumCare Online Portal.
3. Enter your user name and password in the box that appears, then click Login.
4. Under Oracle Support Tools, click Case Manager.
5. In the next screen, click Find Answers to gain access to the Knowledge Base.

SECTION I: INTRODUCING NETPOINT ADMINISTRATION

1

Preparing for NetPoint Administration

Before configuring and administering NetPoint, you may find it useful to preview the tasks that you perform as an administrator. It may also be useful to log in to NetPoint and view the user interface for the COREid System and Access System.

This chapter contains information you need before starting to configure and administer NetPoint, including these topics:

- “Prerequisites” on page 23
- “About COREid System Configuration and Administration” on page 24
- “About Access System Configuration and Administration” on page 29
- “Introduction to Using NetPoint” on page 33

Note: While the NetPoint name is changing to COREid™, in manuals and within the product itself you will continue to see the name NetPoint. NetPoint SAML Services have been renamed to SHAREid and are discussed in the Oblix *SHAREid Administration Guide*.

Prerequisites

NetPoint 7.0 should be installed and set up, as described in the *NetPoint 7.0 Installation Guide*. Read the *Introduction to NetPoint 7.0* manual which provides an overview of NetPoint not found in other manuals.

This volume focuses on COREid System administration as well as common configuration and administration tasks. Included is an introduction to Access System configuration and administration.

About COREid System Configuration and Administration

You use the COREid System to manage identity information about individuals, groups, organizations, and other objects. The Master Administrator of the COREid System can delegate authority to other administrators, allowing the COREid System to scale millions of users.

In addition to managing identity information, the COREid System enables you to manage read, write, and modify privileges for a user based on a specific user attribute, membership in a group, or association with an organization. You can link privileges together into a workflow.

For example, you can set up a self-registration workflow so that when a user self-registers, the registration request is forwarded to appropriate people for approval, and upon approval, the user is immediately and automatically granted access to all resources appropriate for his or her identity attributes.

Finally, the COREid System enables you to accurately manage user identities, group memberships, and organizational objects. This information can then be leveraged by the NetPoint Access System to manage access privileges for users based on user attributes, group membership, or association with an organizational entity.

COREid System Components

The COREid System consists of these components:

- The COREid Server
- WebPass

COREid Server—The COREid Server is a stand-alone server or several instances that manage identity information about users, groups, organizations, and other objects. The COREid Server provides the following applications:

- *User Manager*—If you are an administrator, the User Manager enables you to add, modify, and delete user identities. User Manager data can be leveraged by the NetPoint Access System to provide users with access privileges based on their directory profiles. The User Manager also has reporting capability.

The User Manager typically enables end users to view other users and to modify their own identity information. The users that a person can view and the identity information that someone can modify depends on the privileges granted by a NetPoint Administrator.

- *Group Manager*—Enables administrators to create or delete groups, and enables users to subscribe or unsubscribe from groups. The Group Manager also has reporting capability.

The Group Manager typically enables end users to view groups and to subscribe to membership in a group. The groups that a person can view and subscription rights are granted by a NetPoint Administrator.

- *Organization Manager*—If you are an administrator, the Organization Manager enables you to create and delete organizations and other objects (such as floor plans and assets) that do not belong in the User Manager or Group Manager. The Organization Manager also has reporting capability.

The Organization Manager enables end users to view organizational entities such as floor plans. The organizational entities that a person can view depend upon the rights granted by a NetPoint Administrator.

- *COREid System Console*—Enables administration and configuration of the COREid System. Using the System Console, you also create NetPoint Administrators and assign the right to delegate administrative tasks.

The COREid Server stores user information on a directory server. The COREid Server keeps the directory current so that the Access Server gets the right information.

WebPass—WebPass is a Web server plug-in that passes information between the Web server and the COREid Server. WebPass can talk to multiple COREid Servers.

Details here include:

- “Review of COREid Installation and Setup” on page 25
- “About Configuring the COREid System” on page 26
- “About Managing the COREid System” on page 29

Review of COREid Installation and Setup

Installation and setup of NetPoint includes the following events:

- At least one COREid Server and one WebPass were installed and the resulting COREid System was set up.
- A transport security mode was chosen to protect communication between the COREid Server and WebPass.

- The COREid Server was configured to communicate with an LDAP directory server.

You are prompted regarding automatic setup of your directory server schema. If you chose not to automatically update your schema, you are prompted to do so manually during configuration. Instructions on manual updates of your directory server schema are provided in this manual.

- Each application installed automatically with the COREid Server received a temporary license.

When you log in to the COREid System, you see a series of tabs on the top navigation bar that match your licensed applications. From these tabs you can configure the look and functionality of the User Manager, Group Manager, and Organization Manager applications.

- Required attributes for the User and Group object classes were set up.

Other attributes may also have been configured.

- At least one NetPoint Administrator was selected.

This is the highest-level administrator. You must have at least one administrator defined to begin working with NetPoint. These are the people who configure NetPoint. The NetPoint Administrator creates lower-level administrators called Master Identity Administrators.

Table 1, “Overview of COREid System Configuration,” on page 27 provides a review of COREid installation and setup.

For more information, see the *NetPoint 7.0 Installation Guide*.

About Configuring the COREid System

The COREid System consists of an administrative console and three end-user applications discussed earlier:

- User Manager
- Group Manager
- Organization Manager
- CoreID System Console to configure end-user applications

People use the COREid end-user applications for tasks such as changing personal information, resetting passwords, adding other users, and looking up organizational information. This identity data originates in your LDAP directory. To configure the COREid applications, you need to know what attributes in the directory you want NetPoint to display, and what attributes you want NetPoint to be able to modify.

After configuring NetPoint to work with data in your directory, you configure the COREid application profile pages. These profile pages display the directory data. For example, you can display a user’s name, title, address, and phone number on a profile page in the User Manager application. You can also improve the efficiency of your organization by using COREid workflows. COREid workflows enable you to automate NetPoint-related activities, for example, creating a user and assigning email and other accounts to that user.

Finally, you use the COREid System to create identity workflows. Identity workflows are definitions for a set of actions and the steps you perform to complete the actions. For instance, you can create workflow definitions for the way new employees are added to your various corporate information systems.

Table 1 provides an overview of configuring the COREid System:

Table 1 Overview of COREid System Configuration

Task. . .	Description. . .	Read. . .
Specify additional structural object classes for the Organization Manager and auxiliary object classes for all applications	<p>During setup, you configure one structural object class each for the COREid User Manager, Group Manager, and Organization Manager.</p> <p>You can define additional structural objects classes for the Organization Manager. For instance, you may want the Organization Manager to display assets.</p> <p>You can also add auxiliary object classes to provide the COREid applications with data.</p>	“About Object Classes” on page 60.
Configure attributes	<p>You can determine what attributes are available to the User, Group, and Organization Manager applications.</p> <p>You also can configure rules for how to display attribute values on a COREid application profile page. For example, you may want employees to be able to select their department name from a drop-down list.</p>	“About Object Class Attributes” on page 71.
Configure User, Group, and Organization application tabs	<p>In the User Manager, you configure what the user sees on the My Identity tab.</p> <p>In the Group Manager, you configure what the user sees on the My Groups tab.</p> <p>In the Organization Manager, you configure what the user sees on the Location tab and, optionally, additional tabs.</p>	“Viewing and Modifying Tab Configuration Information” on page 101.

Table 1 Overview of COREid System Configuration

Task . . .	Description . . .	Read . . .
Configure User, Group, and Organization profile pages	<p>Tabs contain one or more profile pages. A profile page contains a set of panels. A panel is a collection of attributes.</p> <p>For example, on a profile page for a user, you can define an Identity panel to display values for attributes such as Name, Photo, Title, and so on.</p>	“Configuring Tab Profile Pages and Panels” on page 113.
Set the searchbase	The searchbase determines the entry point in the directory tree for a search.	“About the Searchbase” on page 126.
Configure view and modify permissions for attributes	<p>You need to determine who can find what, at what point in the searchbase, and with what filter.</p> <p>These decisions affect who can read or write to data and who receives email notification when an attribute has been modified.</p>	“Allowing Users to View and Change LDAP Data” on page 126.
Define workflows	<p>A workflow is a series of steps for creating, deleting, and modifying attributes in COREid.</p> <p>For example, in the User Manager, you may want to define a workflow for creating a user that includes collecting information about the new user from several people in your organization.</p>	“Chaining COREid Functions Into Workflows” on page 171.
Configure password policies	You can determine the length of passwords, frequency of password change, and so on.	“Configuring Password Policies” on page 320.
Delegate administration	To scale your NetPoint installation, you need multiple administrators, each overseeing a subset of NetPoint users.	“Specifying COREid System Administrators” on page 43.

About Managing the COREid System

You can extend your COREid System by adding servers, and expanding your network of COREid System administrators. You can configure audits and logs and perform other administrative functions. Table 2 provides an overview of managing the COREid System.:

Table 2 What to Read for More Information on the COREid System

To perform this task. . .	Read. . .
Add more COREid Servers	<i>NetPoint 7.0 Installation Guide</i> . To ease this process, you may choose to add more COREid Servers through silent installation or cloning, as described in the installation manual.
Add more WebPasses	<i>NetPoint 7.0 Installation Guide</i> . To ease this process, you may choose to add more WebPasses through silent installation or cloning, as described in the installation manual.
Add other COREid System components	<i>NetPoint 7.0 Installation Guide</i> describes how to install most components. For the NetPoint IDLink product, see the <i>Obliv IDLink 1.0 Guide</i> . Information on how to install the Access Server SDK is located in the <i>NetPoint 7.0 Developer Guide</i> .
Configure container limits for Organization Manager	“Setting Container Limits in Organization Manager” on page 165.

About Access System Configuration and Administration

The NetPoint Access System provides centralized authentication, authorization, and auditing to enable single sign-on and secure access control across enterprise resources. You use the Access System to set up security policies that control access to resources. Resources include content, applications, services, and objects in applications on the Web, and similar types of data in non-Web (non-HTTP) resources.

The Access System stores information about configuration settings and access policies in a directory server that uses NetPoint-specific object classes. You can use the same directory to store the Access System configuration settings, access policy data, and the COREid user data, or this data can be stored on separate directory servers.

Access System Components

The Access System consists of the following components:

Access Manager—The Access Manager is installed on a Web server in the same directory as the COREid System component WebPass. See the *Introduction to NetPoint 7.0* manual for an illustration that shows the location of WebPass. The Access Manager provides a login interface to the Access System. Master Access Administrators and Delegated Access Administrators use the Access Manager to define resources to be protected, and to group resources into policy domains. A policy domain consists of resource types to protect, rules for protection, policies for protection, and administrative rights.

The Access Manager has a component called the *Access System Console*, that permits administrators to add, change, and remove Access Clients and Access Servers, configure authentication and authorization schemes, configure master audit settings, and configure host identifiers.

Access Server—The Access Server is a stand-alone server, or several instances, that provide authentication, authorization, and auditing services. The Access Server validates credentials, authorizes users, and manages user sessions. The Access Server receives requests from an Access Client and queries authentication, authorization, and auditing rules in the directory server as follows:

- Authentication involves determining what authentication method is required for a resource, gathering credentials over HTTP, and returning an HTTP response that is based on the results of credential validation.
- Authorization involves granting access based on a policy and an identity established during authentication.

WebGate—The WebGate is an out-of-the-box Access Client for HTTP-based resources. WebGate is an NSAPI or ISAPI plug-in that intercepts HTTP requests for Web resources and forwards them to the Access Server.

The Access System supports single sign-on, enabling you to establish login policies that allow users to access multiple applications with a single login.

For more information, see the topics below:

- “Review of Access System Installation and Setup” on page 31
- “About Configuring the Access System” on page 31
- “About Managing the Access System” on page 33

Review of Access System Installation and Setup

During installation and setup, the following Access System configuration tasks are completed:

- The Access Manager application was installed and configured.
- A directory to store access policies was selected.
- Access Manager was configured to communicate with the directory server that stores access policies.
- One or more authentication schemes may have been configured. Configuring authentication schemes during setup is optional.
- At least one Access Server and one AccessGate were installed and configured.
- The Access Server's transport security communication mode was selected.

Table 3 provides a review of Access System installation and setup, which is described in detail in the *NetPoint 7.0 Installation Guide*.

Table 3 Overview of Access System Installation and Setup

To perform this task. . .	Read. . .
Install the Access Manager	<i>NetPoint 7.0 Installation Guide</i>
Set up the Access Manager	<i>NetPoint 7.0 Installation Guide</i>
Install the Access Server	<i>NetPoint 7.0 Installation Guide</i>
Install a WebGate	<i>NetPoint 7.0 Installation Guide</i>

About Configuring the Access System

The Access System enables you to control who is allowed to access data. You can create access policies beyond the NetPoint applications. For example, if you have an online benefits system, you can configure access policies that only permit employees to view portions of the benefits Web site that are relevant to them. Or you can configure access policies so that external customers are allowed to see your inventory Web pages but not other corporate information.

You do not need to configure the Access Manager application the way you do the COREid Managers. The Access Manager is an interface intended for use only by NetPoint Access Administrators who need to configure access controls.

Note: Table 4 provides an overview of configuring the Access System. For details, see topics in the *NetPoint 7.0 Administration Guide Volume 2*.

Table 4 Overview of Access System Configuration Tasks in Volume 2

Task. . .	Description. . .	Read Volume 2
Enter host IDs	Map host name variations to a single Web server instance.	"Using Host Identifiers"
Create an authentication scheme	Define the method of authentication (for instance, x.509 certificates), the plug-in used to map authentication credentials to a user's identity in the directory, and mapping to the user's DN in the directory.	"Configuring User Authentication"
Create an authorization scheme	Allow the Access Server to make outbound calls to external business logic to determine authorization privileges and actions.	"Configuring User Authorization"
Create a master audit rule	The Access System must have a Master Audit Rule to begin adding data to the audit log file. The audit log file records administrative events such as clearing data from caches.	"Configuring the Master Audit Rule"
Create a policy domain and define resources to protect	A resource is something you want to protect, such as a Web page, plus the actions applied to that item, for instance, an update. A policy domain is a logical set of resources identified by fully qualified path names or URLs that you want to protect, plus the rules for protection, policies for protection, and administrative rights.	"About Policy Domains and Their Policies"
Create policies for URL patterns	Default rules apply blanket coverage for all of the URLs in a policy domain. You can, however, specify individual policies with their own authorization, authentication, and auditing rules for URL patterns and functions such as HTTP get, put, and so on.	"How Are URL Patterns Used?"
Configure single sign-on	Single sign-on allows users to authenticate to multiple applications with one login.	"Configuring Single Sign-On"
Create a shared secret	The shared secret is used to generate the key that encrypts cookies sent between the WebGate and the user's browser.	"Creating a Shared Secret Key"

About Managing the Access System

You manage the Access System by adding more servers, by defining caching parameters, and by extending your access policies using custom plug-ins. Table 5 provides an overview of managing the Access System.

Table 5 Overview of Managing the Access System

To perform this task. . .	Read. . .
Add Access Servers	<i>NetPoint 7.0 Installation Guide</i> . To ease this process, you may choose to add more Access Servers via silent installation or cloning, as described in the installation manual.
Install Access Server SDK	<i>NetPoint 7.0 Developer Guide</i>
Add non-HTTP Access Clients	<i>NetPoint 7.0 Developer Guide</i>
Manage Caching	<i>NetPoint 7.0 Developer Guide</i>

Introduction to Using NetPoint

Before starting to configure NetPoint, it is useful to familiarize yourself with the basics of the product user interface. Commonly used NetPoint functions include the following:

- Login
- Password management
- Navigation bars
- Help
- Selector
- Logout

About Login

NetPoint logs people in based on the roles they have been assigned. As described in “Specifying COREid System Administrators” on page 43, you can specify the following roles for NetPoint users:

- **End User**—An end user can perform searches, view profile data, and modify profile data, depending on access permissions set for individual attributes.
- **Delegated Administrator**—A Delegated Administrator is a user who can perform all of the same tasks as an end user and can create user, group and

organization objects, depending on the level of permissions he or she has been granted. A Delegated Administrator can also view requests.

- **Delegated Identity Administrator**—A Delegated Identity Administrator is a user who has been delegated the right to view configuration tabs for the User Manager, Group Manager, and Organization Manager applications. This person can set attribute access controls, define workflows, and so on.
- **NetPoint Administrator**—A NetPoint Administrator can view the User Manager, Group Manager, and Organization Manager applications, and use COREid System configuration functions in the System Console.

For example, if you log in to NetPoint as a NetPoint Administrator, you can view every screen in every application. But if you log in as an end user, you may only see a subset of the User, Group, and Organization Manager applications, and you cannot access COREid administrative functions.

By default, NetPoint offers single sign-on between the COREid and Access Systems. If you log in to one system, you should not be prompted to log in to the other system.

If you use the Access System to protect the NetPoint applications, you can bypass the default login form and implement your own custom form. For details about protecting resources with policy domains, see the *NetPoint 7.0 Administration Guide Volume 2*.

Logging In to the COREid System

The procedure for logging in to the NetPoint COREid System depends on whether you customized the login screen, made it available as a portal insert, or protected it with the NetPoint Access System. This section covers the default login screen that ships with the NetPoint COREid System, as well as the impact of the default user type on login. See the *NetPoint 7.0 Customization Guide* and the *NetPoint 7.0 Developer Guide* for more information on customization.

You must configure an attribute with a semantic type of Login before users can log into the COREid System. You can either automatically configure this attribute during installation, or manually configure it from the NetPoint COREid System Console. See “Making Schema Data Available to NetPoint” on page 59 for more information.

Note: Only NetPoint Administrators and Master Identity Administrators have access to the COREid System Console. See “Specifying COREid System Administrators” on page 43 for more information about configuring these administrators.

To log in to the COREid System

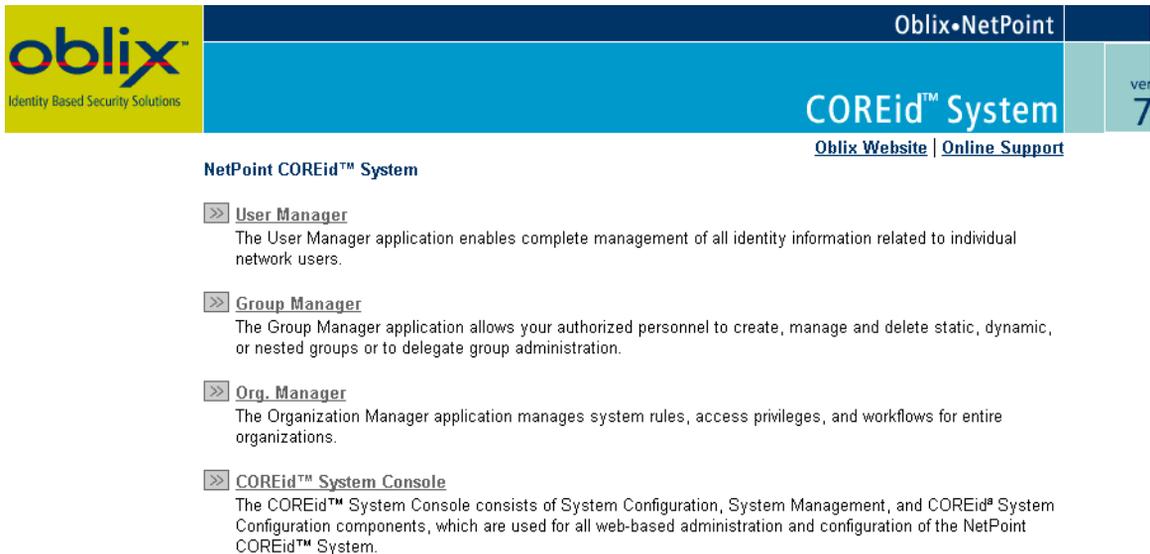
1. In your browser, type the path to the COREid System and press Return.

For example:

`https://hostname:port/identity/oblix`

where *hostname* is the name of the computer on which the WebPass is installed and *port* is the Web server port for the WebPass. You can log in using the HTTP or HTTPS protocol.

By default, the following screen appears:



Oblix•NetPoint

Oblix
Identity Based Security Solutions

COREid™ System ver 7

[Oblix Website](#) | [Online Support](#)

NetPoint COREid™ System

- >> **User Manager**
The User Manager application enables complete management of all identity information related to individual network users.
- >> **Group Manager**
The Group Manager application allows your authorized personnel to create, manage and delete static, dynamic, or nested groups or to delegate group administration.
- >> **Org. Manager**
The Organization Manager application manages system rules, access privileges, and workflows for entire organizations.
- >> **COREid™ System Console**
The COREid™ System Console consists of System Configuration, System Management, and COREid® System Configuration components, which are used for all web-based administration and configuration of the NetPoint COREid™ System.

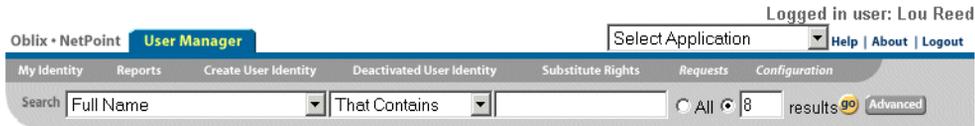
2. Select the desired application.

The login screen appears.

3. Enter your user name and password.
4. For Active Directory users, if the Domain field is present, select the domain in which this installation of the COREid System operates.

5. Click Login.

By default, when you log in to the COREid System application, you see all of the functions available to a NetPoint Administrator. For example, in the User Manager, you will see functions such as “My Identity,” “Reports,” and the search function.



See the *NetPoint 7.0 Customization Guide* for more information about changing this default.

Logging In to the Access System

By default, NetPoint does not require a user to log in to the Access System if he or she is already logged in to the COREid System, and vice versa. Session information is stored in a cookie called the ObTEMC cookie. You may choose to protect the NetPoint applications in a policy domain, in which case a different authentication can be used. For details about protecting resources with policy domains, see the *NetPoint 7.0 Administration Guide Volume 2*.

You must configure an attribute with a semantic type of Login before users can log into the Access System. You can either automatically configure this attribute during installation, or manually configure it from the NetPoint COREid System Console. See “Making Schema Data Available to NetPoint” on page 59 for more information.

This section covers the default login screen that ships with the NetPoint Access System.

Note: Only NetPoint Administrators and Master Access Administrators have access to the Access System Console. For details about configuring Master Access Administrators, see the *NetPoint 7.0 Administration Guide Volume 2*.

To log in to the Access System

1. In your browser, type the path to Access System and press Return.

Example: `https://hostname:port/access/obliv`

Hostname is the name of the computer on which the Access Manager is installed. *Port* is the Web server port for the Access Manager. You can log in with the HTTP or HTTPS protocol.

The following screen appears.

NetPoint Access System™

» **Access Manager**

The Access Manager application allows you to create, remove and manage policies and resources and test policy enforcement.

» **Access System™ Console**

The Access System™ Console consists of System Configuration, System Management, and Access System Configuration components, which are used for all web-based administration, and configuration of the NetPoint Access System™.

NetPoint COREid™ System

The NetPoint COREid™ System consists of the User Manager, Group Manager and Organization Manager applications and the COREid™ System Console.

2. Select the application you want.

The login screen appears.

3. Enter your user name and password, then click the Login button.

Access Manager—Only Delegated Access Administrators will actually see any Policy Domains. Otherwise, a screen similar to the following appears. For details about delegating administration in the Access Manager, see the *NetPoint 7.0 Administration Guide Volume 2*.

Logged in user: Christopher Yen

Oblix • NetPoint

Access Manager

Search
My Policy Domains
Access Tester
Help
About
Logout

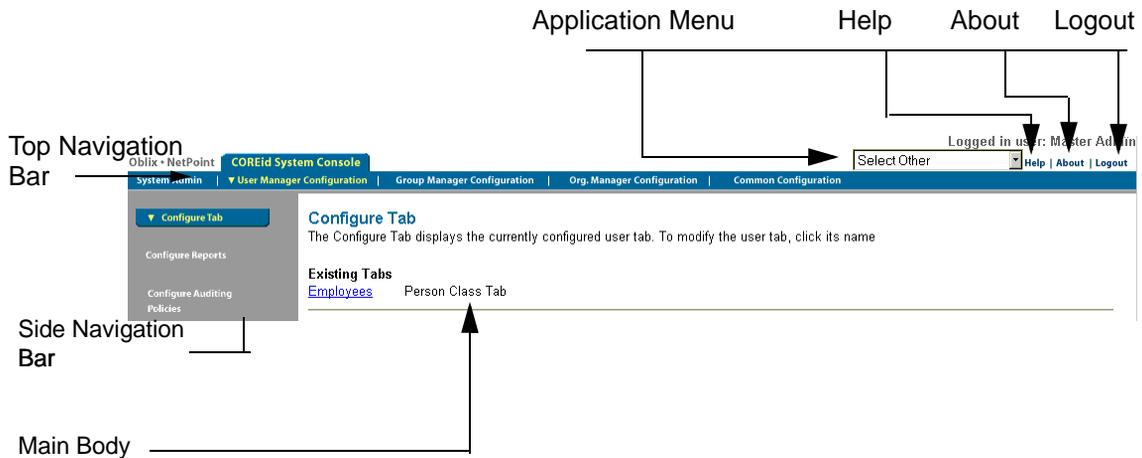
My Policy Domains

There are no policy domains defined.

Access System Console—Only NetPoint Administrators and Master Access Administrators can access its functions. For more information about configuring Master Access Administrators, see the *NetPoint 7.0 Administration Guide Volume 2*.

Screen Functional Areas

The following is a COREid System Console screen that appears after login when you select the User Manager Configuration tab in the top navigation bar, and the Configure Tab item in the side navigation bar.



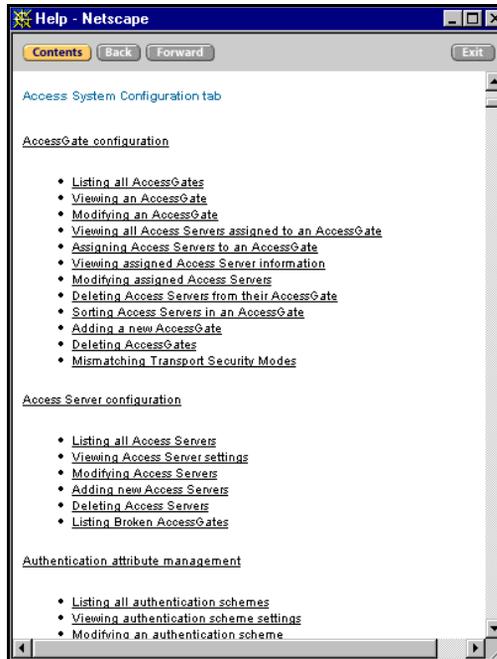
All NetPoint screens have the following functional areas:

- **Application Menu**—A drop-down list for accessing NetPoint applications.
- **Top Navigation Bar**—Links to applications and functions are also available at the top of application and System Console screens.
- **Side Navigation Bar**—The side navigation bar contains the links that are available for the function selected in the top navigation bar.
- **Main Body**—The main body displays a description of the currently selected function or the fields to be completed.

Online Help

A Help link is located at the top right of COREid System screens, and in the side navigation bar of Access System screens. To access online Help, click this link.

An example of a NetPoint Help screen is shown below.



You can perform the following tasks in the online Help window:

- Scroll to view the entire Help topic.
- Click Contents to display a list of topics.
- Click Back or Forward to see other Help topics.
- Click Exit to close the window.

The About Link

An About link is located at the top of COREid System screens, and in the side navigation bar of Access System screens. Click About to display the Oblix address, telephone numbers, and other contact information, and copyright information.

These buttons appear on the About page:

- **View System Info**—Displays the server platform, version of NetPoint installed, and product licenses installed.
- **Submit Admin Feedback**—Displays a form that lets you email a message to an internal administrator.
- **Submit Oblix Feedback**—Displays a form that lets you email a message to Oblix.
- **View License**—Displays the license agreement.

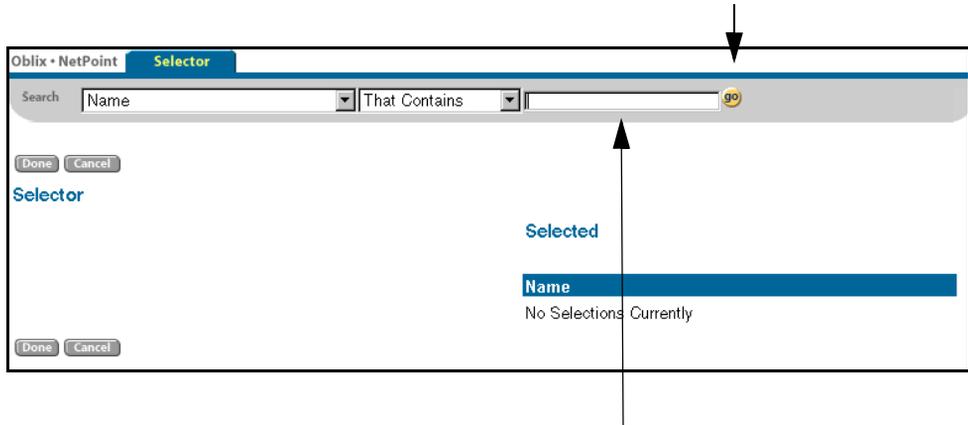
The Selector

You use the Selector to search for users or groups. The Selector is available when you view, create, or modify a profile or a workflow. To invoke the selector, click the Select User or Select Group button.



To use the Selector

1. Click Select User or Select Group to display the Selector screen.



2. For the simplest kind of search, type a text string of at least three characters in the right-most field.
3. Click Go.

Users or groups matching your search criteria appear on the screen.

NetPoint displays 8 hits per page. This applies to both Selector and Query Builder. If you perform a search and/or query that results in more than 20 hits, you receive truncated results. For instructions on changing this search cap, refer to the `cookieBustLimit` parameter documentation in the *NetPoint 7.0 Customization Guide*.

4. Click users or groups to select them.

The selected names appear under Selected, on the right side of the screen. You can modify the list by clicking the checkbox of an entry. One click deselects an entry; a second click reselects it.

5. Click Done.

The selected names appear in the NetPoint window from which you invoked the Selector.

Note: If you receive a “Bad request” message when you click Done, your search string is too long for your browser. Browsers handle the search parameters as URLs, and they generate an error if the search exceeds their maximum URL length.

Logging Out

A Logout link is located at the top of COREid System screens, and in the side navigation bar of Access System screens. By default, if you log out of the COREid System, you are automatically logged out of the Access System and vice versa.

When you finish using NetPoint to prevent unauthorized people from accessing your information, you should log out and close your browser.

By default, sessions expire after three hours. To change the timeout, see “Configuring Session Timeout” on page 278 for details.

To log out of the COREid System

1. Click Logout in the upper right-hand corner.
2. Click OK when prompted to close your browser.

To log out of the Access System, from any Access System application click Logout in the side navigation bar.

The next chapter explains how to specify COREid System administrators.

2 Specifying COREid System Administrators

This chapter explains how to specify administrators for the COREid System.

This chapter contains the following topics:

- “About COREid System Administrators” on page 43
- “Specifying Administrators” on page 46
- “Delegating Administration” on page 47

About COREid System Administrators

The NetPoint COREid System manages user, group, and organization identity information, as described in the *Introduction to NetPoint 7.0* manual.

Administering the COREid System involves a broad range of tasks that are designed to help you manage your data, enhance performance, and control the appearance and functionality of COREid applications. For details about these tasks, see “Configuring and Managing the COREid System” on page 265.

The responsibility of administering the COREid System is shared between NetPoint Administrators and Master Identity Administrators:

NetPoint Administrators—At least one NetPoint Administrator is specified when NetPoint is set up. This is the highest level administrator in NetPoint. A NetPoint Administrator can specify other NetPoint Administrators and Master Identity Administrators.

Master Identity Administrators—Master Identity Administrators can delegate specific responsibilities to administrators called Delegated Identity Administrators.

Delegated Identity Administrators—Delegated Identity Administrators are Assigned by Master Identity Administrators and created in User Manager.

See Table 6 for a description of the types of COREid System Administrators and their privileges.

Table 6 Types of COREid System Administrators

Administrator	Becomes an Administrator When	Tasks Performed
NetPoint Administrator	Assigned when NetPoint is installed	<ul style="list-style-type: none"> • Assigns other NetPoint Administrators and Master Identity Administrators • Assigns self as a Master Identity Administrator • Manages all System Configuration and System Management functions of the COREid System Console • Configures COREid Server • Specifies administrators • Configures styles • Configures directory server profiles • Configures WebPass • Configures Password policies • Manages COREid Server settings • Imports photos • Manages log files and audit files

Table 6 Types of COREid System Administrators

Administrator	Becomes an Administrator When	Tasks Performed
<p>Master Identity Administrator</p>	<p>Assigned by NetPoint Administrator</p>	<ul style="list-style-type: none"> • Assigns Delegated Identity Administrators • Manages all three COREid applications: User Manager, Group Manager, and Organization Manager • Manages Common Configuration as well as application-specific configurations in the COREid System Console <p>Common Configuration Tasks:</p> <ul style="list-style-type: none"> • Configures object classes • Configures workflow panels • Configures master audit policies • Configures logging and auditing policies <p>User Manager Configuration Tasks:</p> <ul style="list-style-type: none"> • Configures tabs • Configures reports • Configures logging and auditing policies <p>Group Manager Configuration Tasks:</p> <ul style="list-style-type: none"> • Configures tabs • Configures reports • Configures group types • Configures Group Manager options • Configures logging and auditing policies <p>Organization Manager Configuration Tasks:</p> <ul style="list-style-type: none"> • Configures tabs • Configures reports • Configures logging and auditing policies

Table 6 Types of COREid System Administrators

Administrator	Becomes an Administrator When	Tasks Performed
Delegated Identity Administrator	Assigned by Master Identity Administrators	<ul style="list-style-type: none"> • Assigns other Delegated Identity Administrators • Manages assigned tasks • Delegates administration • Configures attribute access control • Defines workflows • Monitors workflow status • Sets searchbase • Expands dynamic groups • Sets container limits

Specifying Administrators

You use the COREid System Console to assign NetPoint and Master Identity Administrators. As mentioned earlier, you must be a NetPoint Administrator to complete this task.

To specify NetPoint and Master Identity Administrators

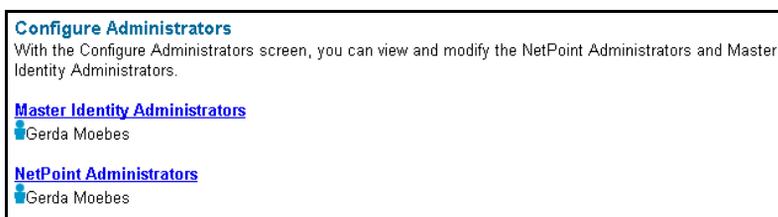
1. Log in to the COREid System Console as a NetPoint Administrator.
2. Click System Configuration.

The System Configuration page appears.

3. Click Configure Admins in the side navigation bar.

The Configure Administrators page appears, displaying two options: Master Identity Administrators and NetPoint Administrators.

See Table 6 for a list of the tasks performed by each type of administrator.



4. Click the category of administrator you want to add.

The Modify *type_of_administrator* page appears.

where, *type_of_administrator* is either a NetPoint Administrator or a Master Identity Administrator.

5. Click Select User to add an administrator.
See “The Selector” on page 40 for information about using this feature.
6. Select a user and click Add.
The name you select in the Selector page appears in the Modify *type_of_administrator* page, where *type_of_administrator* is a NetPoint Administrator or a Master Identity Administrator. You can specify multiple administrators.
7. Click Done to leave the Selector page.
8. Click Save to add the administrator, or click Cancel to exit without saving your changes.

Deleting Administrators

When you delete an administrator, you remove administration rights from the user, but you do not remove or deactivate the user from the LDAP directory.

To delete an administrator

1. In the Configure Administrators page, click the category from which you want to delete one or more administrators.

The Modify *type of administrator* page appears, where *type of administrator* is a NetPoint Administrator or a Master Identity Administrator.

2. Clear the check box next to the administrator you want to delete.
3. Click Done to confirm the deletion, or click Cancel to stop the deletion.

Delegating Administration

You can delegate your rights and responsibilities to other administrators. The tasks delegated are specific to the delegated right, the target, and the tree path.

This section covers:

- “About Delegating Administration” on page 48
- “Delegated Administration Models” on page 49
- “Adding Delegated Administrators” on page 52

About Delegating Administration

Delegating administration allows the NetPoint Administrator and Master Identity Administrator to delegate their responsibilities to other, more local administrators. This is particularly useful in large organizations, where it may be necessary to administer thousands or millions of users.

When you delegate administration, you determine what rights you want to grant to another user. Rights include the ability to configure the following:

- Read access for attributes
- Write access for attributes
- Notification by email of attribute modifications
- Setting the searchbase
- Monitoring requests
- Defining workflows
- Containment limits

In addition, you can designate people to act as your substitute. People who are granted substitution rights can temporarily perform any of the functions that you are permitted to perform.

After you have delegated a right to another user, that user becomes a Delegated Identity Administrator. By delegating administration, you determine who can configure or access which feature, at what level, and with which filters. Configuration or access authority may be for a specific user or group of users, a role, or a rule. The resource that can be configured or accessed may include a searchbase, an attribute access control, a workflow definition and so forth. The level is the starting DN.

Task overview: Delegating administrators

1. Start the delegation procedure for the desired application.

Note: All activities here are described in “Adding Delegated Administrators” on page 52.

2. Select the right that you want to grant (for Read, Write, and Notify permissions only).
3. Identify the attribute associated with the right.
4. Specify the level of access control for that attribute, thus setting the scope of the directory tree to which the rights apply.
5. Select the person to whom you are delegating the rights.

For example, as the Master Identity Administrator, you can grant one or more users the ability to set Read access control for the Title attribute. You can specify whether you want the Delegated Identity Administrator to be able to delegate access control to others.

For more information, see “Adding Delegated Administrators” on page 52.

Delegated Administration Models

The COREid System enables you to set access controls and delegate administration for directory tree structures that represent different business models. These models include an extranet model, an intranet model, and an ASP model. These models are described in the following sections:

- “Extranet Model” on page 49
- “Intranet Model” on page 50
- “ASP Model” on page 51

Extranet Model

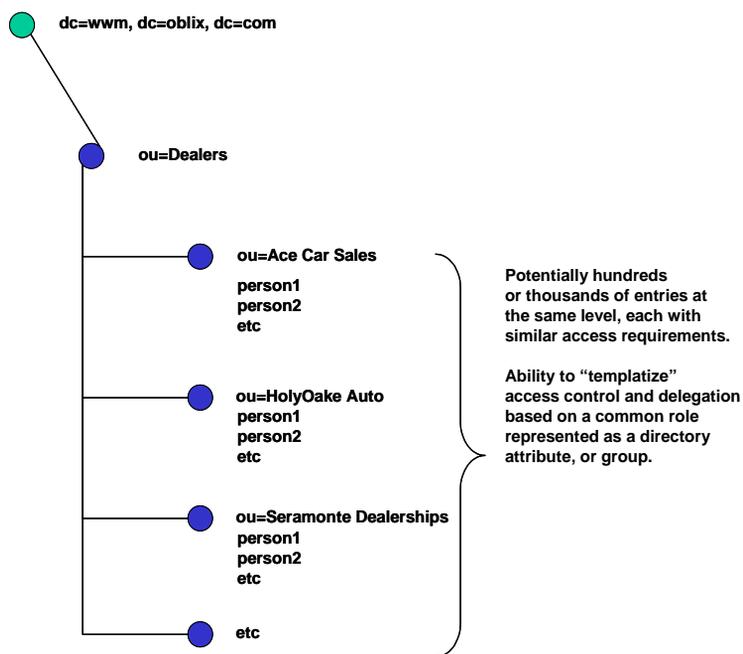
A typical business-to-business extranet might have 500 or more extranet organizations using a site. These organizations represent customers, partners, and suppliers, each having between 1 and 100 users.

The goal in the extranet model is to have the Master Identity Administrators push out administrative responsibilities to each of the partners. But because there are so many partners, it would be a burden to define new roles and responsibilities each time a partner joined. Therefore, the Directory Administrator must define a fixed set of roles and responsibilities that are leveraged across all customers, existing and new. The Master Identity Administrator can then define access controls and create delegated administrator policies that are symmetric across all organizations.

The Delegated Identity Administrator at each partner site is typically a line-of-business person who has a fixed, well-defined set of tasks and rights, such as creating users and changing attribute access permissions. Delegated Identity Administrators can only give others in their organization administrative privileges by adding and deleting people from a set of pre-defined roles.

For example, the Delegated Identity Administrator creates a new user with an attribute of `admin=yes`. This new user then inherits the ability to change attribute access control permissions, create new users, and other well-defined tasks, as illustrated in Figure 1.

Figure 1 Extranet Delegated Administration Example



Intranet Model

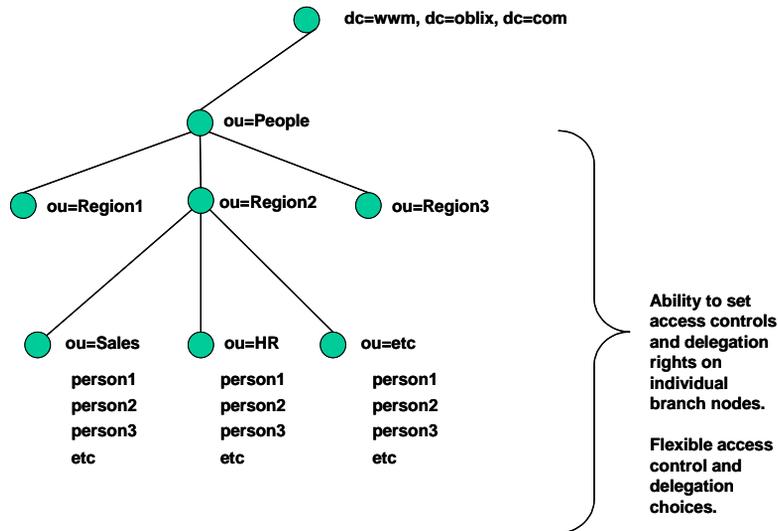
In a typical intranet model, the directory tree is generally organized according to a logical separation of users, such as by geography (North America and Europe) or function (Marketing and Engineering).

The directory might be characterized by only a few branches at each OU, but may be several layers deep in branching. The branches may be very different from each other and may have several thousand users in each branch. At a given node, a European branch might have 500 users under Sales and Marketing, while a North American branch might have 10,000 users under East, Central, and West.

The Master Identity Administrator may choose to delegate administration centrally or at the OU level, depending on where the technical and business process knowledge resides. Or additionally, the Master Identity Administrator may choose to delegate administration across specific tasks; for example, you might delegate the task of provisioning phone numbers—but not managing access permissions or creating new users.

Figure 2 illustrates the intranet model:

Figure 2 Intranet Delegated Administration Model



ASP Model

Some business-to-business extranet sites may follow the application service provider (ASP) model more closely than the extranet model described earlier.

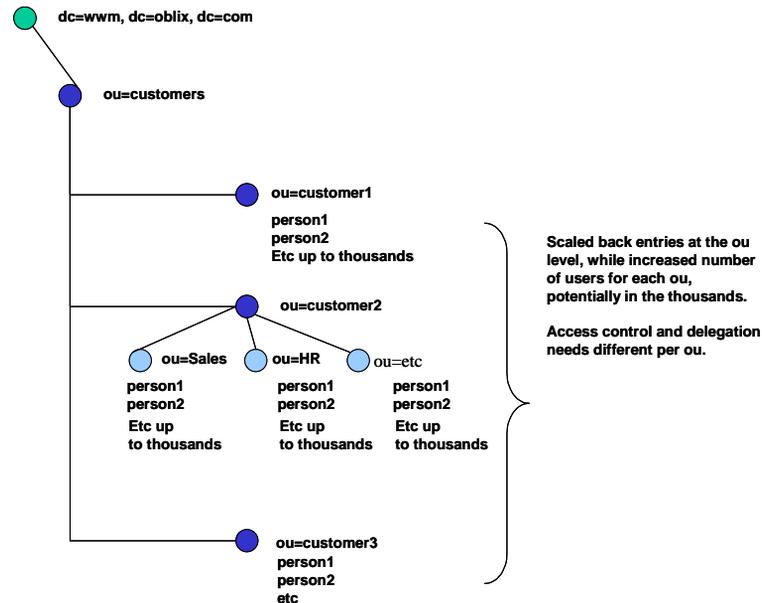
In an ASP model, there are fewer extranet partners but significantly more users at each partner site. For example, there may be only approximately 50 partners but each partner may have 1000 users.

ASPs provide hosted services. Different customers may need different sets of services. This means the scope of data that needs to be managed, such as access rights, may differ for each OU. Further, the directory structure of each OU may be substantially different. Under each OU may be all the complexity of an intranet directory tree as in the intranet model, yet the structure of the tree could be completely different between the OU for Customer 1 and the OU for Customer 2.

The ASP model needs a flexible delegation model similar to the intranet model. The Master Identity Administrators at the ASP site performs some top-level configuration, such as setting the searchbase, and configures an initial delegation model similar to the extranet model. However, each customer site requires the flexibility to create a customized delegated administration model, either by a technical Delegated Identity Administrator at the customer site or by the Master Identity Administrators at the ASP site.

Figure 3 illustrates the ASP model.

Figure 3 ASP Delegated Administration Model



Adding Delegated Administrators

Delegating administration allows the Master Identity Administrator or a Delegated Identity Administrator to further delegate responsibility to other local administrators.

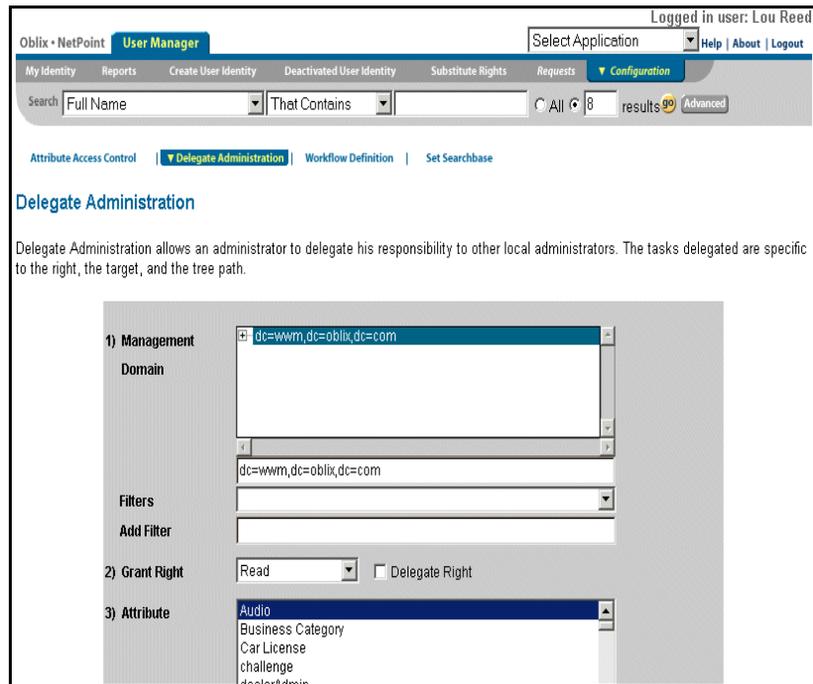
To delegate administration

1. Log in to the User Manager, Group Manager, or Organization Manager.
2. Click the Configuration option at the top of the page.

The Configuration page appears.

3. Click Delegate Administration.

The Delegate Administration page appears.



4. In the Management Domain box, specify the scope of the DIT that this right applies to.

Initially this field displays the searchbase defined during setup. The searchbase is usually defined at the highest (company-wide) level. Depending on the level of your delegated rights, you can specify access control at any level, from the lowest level (an individual user), through intermediate levels (departments, divisions, partners), and then to the highest level (company-wide). For example, if you select the Full Name attribute and select a department such as Sales, you are setting an access control that applies to all full names belonging to the Sales department.

The field below the Management Domain box displays the selection.

5. Optionally, use the Filters field to specify either a variable substitution or LDAP rule to filter the DIT level you selected.

For more information, see “Usage of Rules and Filters” on page 89.

6. Optionally, in the Add Filter field, enter another filter, then click Save.

The new filter displays in a field below the previous filter.

7. In the Grant Right drop-down list, select the right that you want to grant to the delegated administrator:

- **Read**—Allowed to set read (view) permission for the selected attribute
 - **Modify**—Allowed to set modify permission for this attribute
 - **Notify**—Allowed to set notify permission when user requests attribute value change
 - **Set Searchbase**—Allowed to specify the searchbase
 - **Monitor Requests**—Allowed to monitor requests and manage deactivated users
 - **Define Workflow**—Allowed to define workflows
 - **Substitute Rights**—Allowed to designate other people as your substitute
8. Give the new administrator the authority to further delegate this right to other administrators by selecting the Delegate Right check box.

Note: Selecting Delegate *does not* automatically assign Grant rights. You must define Delegate and Grant rights separately.

9. In the Attribute box, select an attribute to associate with the right.
10. Select a trustee to whom you want to assign one or more rights with one or more of the following methods:
- **Rule**—Click Build Filter and use the Query Builder to create a rule. See “Writing LDAP Filters Using Query Builder” on page 132 for details.
 - **Person(s)**—Click Select User and use the Selector to specify one or more users.
 - **Group(s)**—Click Select Group and use the Selector to specify one or more groups.
- The Rule, Person(s), and Group(s) fields have an *or* relationship. A user specified in any of these fields is assigned the right.
11. Use the Copy and Paste buttons to copy users and groups from one attribute to another.
- Click Copy, click Reset, select another attribute, and click Paste. The users and groups appear in their respective boxes.
12. Click one of these buttons:
- **Save**—Saves and implements your changes
 - **Reset**—Clears all selections
 - **Delete**—Clears all rule, group, and user specifications
 - **Report**—Generates a report of all attributes and their access permissions across the domain

Adding Substitute Administrators

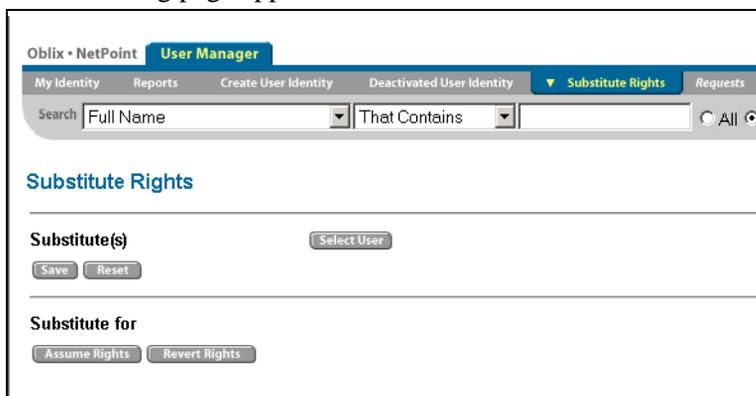
As a COREid Administrator, if you have been granted substitute rights, you can designate other people to temporarily assume your rights. After your substitute logs into NetPoint COREid, they can assume your identity. When your substitute views the My Identity page, your information is shown rather than the other person's information.

By assigning substitute rights, you allow someone else temporarily to assume your identity. For example, suppose you are a Delegated Identity Administrator. Before leaving for vacation you assign substitute rights to J. Smith. When J. Smith logs in, he assumes your identity. Later, when J. Smith wants to perform his own duties, he reverts the delegated rights. Although the substitute appears to be you while assuming your identity, NetPoint logs all activities with both the substitute's and your identities. All logs and alarms show duplicate entries using both identities.

To assign or remove a substitute

1. Log in to the User Manager and select Substitute Rights.

The following page appears:



The screenshot shows the 'User Manager' interface for 'Obliv • NetPoint'. The navigation bar includes 'My Identity', 'Reports', 'Create User Identity', 'Deactivated User Identity', 'Substitute Rights' (selected), and 'Requests'. A search bar contains 'Full Name' and 'That Contains'. Below the search bar, the 'Substitute Rights' section has a 'Substitute(s)' field with a 'Select User' button and 'Save' and 'Reset' buttons. The 'Substitute for' section has 'Assume Rights' and 'Revert Rights' buttons.

If you have been granted substitute rights, this page contains a Select User button. If you have designated people to be your substitute, these people are listed in the Substitute(s) field. This page also contains a Substitute for field with a list of people who have designated you as their substitute. If no people are listed, no one has designated you as their substitute.

2. Assuming that you have been granted substitute rights, click Select User.

The Selector page appears.

3. Select the user and click Add.

The user is added to the Selected list.

4. Select a user and click Delete to remove the user.

The user is removed from the Selected list.

5. Click Done to leave the Selector page.
6. Click Save to save your changes.

To assume an identity

1. Log in to the User Manager and select Substitute Rights.
2. Select the user whose rights you wish to assume.
This user must already have assigned you to be a substitute.
3. Select Assume and Save.

To revert to your own identity

1. From the User Manager, select Substitute Rights.
2. Select Revert.

See “Configuring and Managing the COREid System” on page 265 for details about configuring styles for COREid applications, configuring multiple languages for NetPoint, configuring and managing COREid Servers and WebPass, and configuring password policies and the Access Server SDK for the COREid System.

SECTION II: CONFIGURING THE COREID SYSTEM

3 Making Schema Data Available to NetPoint

The COREid applications (the User, Group, and Organization Manager) enable users to view and modify data about themselves, other users, groups, and other objects. The items that users see on the COREid applications consist of LDAP directory attributes that you have configured in the COREid System Console. In order for the COREid applications to display data, you use the COREid System Console to configure objects and attributes from a directory schema that the application is to work with.

The COREid applications also enable users to provision accounts in back-end applications. For example, users can enter data that creates new user accounts in Exchange through the MIIS provisioning application. To prepare the COREid applications to be used for provisioning, you use the COREid System Console and configure objects and attributes from a template schema. A generic schema file is supplied with the COREid System.

This chapter discusses the following topics:

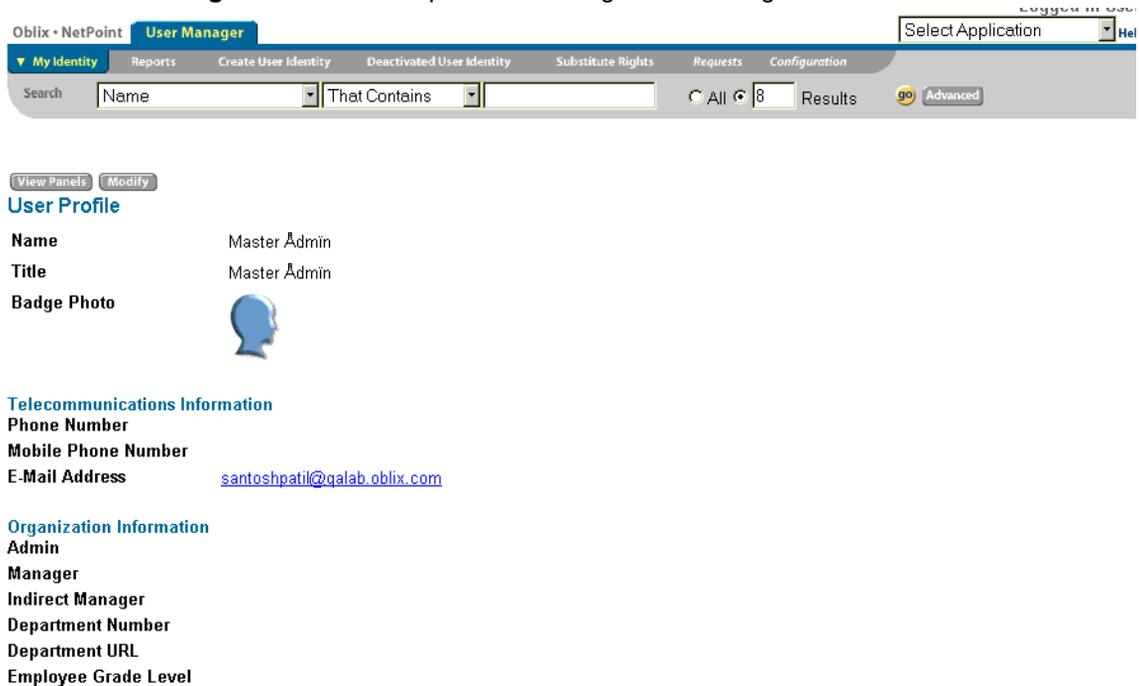
- “About Object Classes” on page 60
- “Viewing Object Classes” on page 65
- “Modifying Object Classes” on page 67
- “Adding Object Classes” on page 69
- “Deleting Object Classes” on page 71
- “About Object Class Attributes” on page 71
- “Viewing Attributes” on page 80
- “Configuring Attributes” on page 81
- “Configuring Derived Attributes” on page 94
- “Attributes Configured on a Per-Application Basis” on page 97

About Object Classes

As a COREid administrator, you configure three applications for your users (and other administrators). These applications are the User Manager, Group Manager, and Organization Manager.

Figure 4 shows a portion of a User Manager Profile page.

Figure 4 Sample User Manager Profile Page



The COREid applications obtain most of the data that they display from entries in an LDAP directory. For instance, the User Manager may show a person's name, email, and so on. This data is taken from attribute values stored on a person object in the directory. These attributes and their values are displayed on profile pages in the User Manager. In Figure 4 on page 60, the name displayed in the user profile is based on the name attribute for the person object in the directory. The actual name being displayed is a value stored with the attribute. The title is based on the title attribute for the person object.

All of the COREid System applications—the User Manager, Group Manager, and Organization Manager—display attribute values for specific objects on profile pages.

About Template Objects and Provisioning

In addition to configuring objects and attributes from an LDAP directory, the COREid System allows you to define template schemas. Using the COREid System Console, you configure the objects and attributes from a template schema in the same way that you configure LDAP data. However, LDAP data and template data are used in different ways. You configure LDAP data to populate COREid applications. Users enter values for LDAP data from either a COREid application profile page or from a workflow. The LDAP data is displayed on the profile page. In contrast, you can only enter template data during a workflow step. The data is not displayed on the profile page. Instead, it is sent to a back-end application to provision application accounts for users.

Unlike LDAP data, you can use only template object data for provisioning. For example, to provision email accounts from a COREid workflow, you would create an object template schema with attributes that the email provisioning system can understand. You would then configure the attributes from this schema in the COREid System, define a workflow that uses these attributes, and use the Identity Event API to send this data to the back-end system.

Note: For instructions on how to configure the template object file, see Chapter 6, “Provisioning External Applications from COREid” on page 253. The template object file must be finalized before you can configure the template objects in the COREid System Console, as described in Chapter 6.

The Process for Configuring Schema Data

When you first install and set up NetPoint, the User Manager, Group Manager, and Organization Manager applications are empty. You need to configure these applications with information. For example, you may want the User Manager to display information about a user such as their name, title, phone number, email, and so on. Before configuring the appearance of these applications, you must first use the COREid System Console to configure the LDAP objects and attributes that you want to display on the application profile page. You must also define how each attribute is to be displayed, for example, if it is a string value, a selection list, or a radio button. You primarily use LDAP directory data to identify in the COREid System Console which objects and attributes you want to display to users on application profile pages.

Once you have configured objects and attributes in the COREid System Console, you can configure COREid applications to display these attributes and their values. Finally, you assign View and Modify rights to determine who can view and modify these attributes. Configuring the COREid applications is discussed in “Configuring User, Group, and Organization Manager” on page 99.

If you are using the COREid System as the entry mechanism for provisioning user application accounts, you use the COREid System Console to configure template objects and attributes. As with LDAP data, you define how each attribute is to be displayed on a profile page. The primary difference between LDAP and template data is that users may only enter values for template attributes during a workflow step, and you must use the Identity Event API to pass this data to a back-end application.

Objects Configured During Installation

During installation and setup of the COREid System, you configured the following object classes:

- **User Manager**—A required person object class
- **Group Manager**—A required group object class
- **Organization Manager**—A NetPoint-provided location object class

These object classes are taken from the LDAP directory that the COREid System communicates with.

In addition to the object classes that are configured during the installation process, you may want to configure additional LDAP and template-based objects and attributes in NetPoint. The following sections discuss how you configure objects and attributes to provide the COREid applications with data.

Note: Configuring an attribute does not ensure that the attribute is displayed on a COREid application page. You must associate specific attributes with a COREid application page, and assign View and Modify rights to these attributes. See “Configuring User, Group, and Organization Manager” on page 99 for details.

Structural and Auxiliary Object Classes in NetPoint

The COREid System works with *structural* and *auxiliary* LDAP object classes. When you install COREid, the User, Group, and Organization Manager applications are associated with one structural object class each. A structural object class describes the basic aspects of an object. Examples of structural object classes include person and groupOfNames. The person object class may contain attributes such as name, department, employee ID, and email address.

The User Manager and Group Manager are always associated with only one structural object class.

The Organization Manager can be associated with any number of structural object classes of a generic or location object class type (see “Object Class Types” on page 64). During installation and setup, a location structural object class is associated with the Organization Manager. In the Organization Manager, each structural

object class is represented as a link at the top of the work area in the user interface. These links in the Organization Manager are referred to as *tabs*, although they resemble links rather than physical tabs.

Figure 5 Organization Manager Tabs

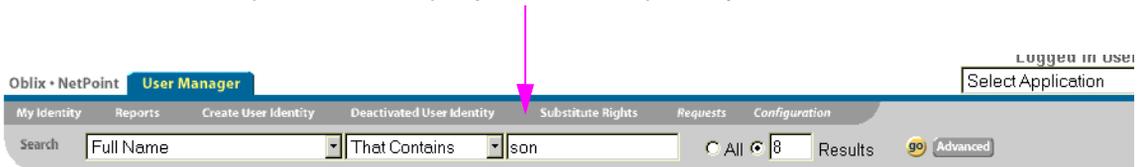
Organization Manager tabs. Each tab represents a structural object class in the directory. Links in the Org. Manager tab are also referred to as tabs.



Container Limits



The User Manager and Group Manager have only one tab, corresponding to the person and Group object classes, respectively.



You use an auxiliary object class to add a set of related attributes to an entry that already belongs to a structural class. Auxiliary object classes are *mix-in* LDAP object classes that can be added to any structural class. Items such as a billing address, a challenge phrase, a response to a challenge phrase, and so on may be useful for definition in an auxiliary object class.

You must configure attributes for each object class that you want to manage through the COREid System Console. See “About Object Class Attributes” on page 71 for more information.

Notes

Before users can see values for attributes that you configure, you must set up the User Manager, Group Manager, and Organization Manager tabs and provide view and modify permissions, as appropriate, to the users. See “Configuring User, Group, and Organization Manager” on page 99 for details.

The inheritance of all objects is based on the premise of a common super class for both the structural object class and the auxiliary class. Otherwise, object class extension is not feasible. For example, if you don't choose Top as the inherited Object Class in eDirectory, NDS sets the inherited object class to None. When you configure the object class as an auxiliary class in the COREid System, no problems are evident initially. However, if you execute a Create User Workflow that contains attributes of this auxiliary class, the Enable step fails when trying to commit because the schema is incompatible and the auxiliary class cannot be added to the entry's objectclass attribute due to a schema violation.

Template Object Classes

In addition to structural and auxiliary object classes, the COREid System recognizes *template* object classes. Template object classes function in part like auxiliary object classes, in that they are used to augment the functionality on a COREid application tab, and they cannot be used as the foundation for a tab. However, template objects are not defined in an LDAP directory, and you do not use template objects for configuring the profile pages shown on a tab.

You define template objects in a schema file. The template objects are only used in COREid workflows, for sending data to back-end applications. Template object classes differ from the other kinds of object class in several respects:

- Users interact with the template object class attributes for the purpose of provisioning accounts in back-end applications.
- Users interact with template object class data attributes only in the context of a COREid workflow.

Template attribute values are only visible when a user invokes a workflow instance and enters the data. Once the data is submitted, it cannot be displayed again in the COREid System. See “Chaining COREid Functions Into Workflows” on page 171 for details. This is different from LDAP data, which you use to configure the fields, labels, and other items displayed on the COREid application pages.

Note: Template attribute values cannot currently be displayed because the flow of data from COREid to the back-end system is one-way. This limitation will be removed in a future release.

- Template object data resides a template object file, not an LDAP directory.

For information on defining template objects and the complete process for using them to provision back-end applications, see “Provisioning External Applications from COREid” on page 253.

Object Class Types

When configuring your object classes in NetPoint, you are asked to specify an object class *type*. The term object class type refers to how an object class is used within the COREid System. The following are object class types.

Type	Description
Person	This type contains information about a person. Examples of this type include <code>companyOrgPerson</code> and <code>customerOrgPerson</code> . When you install NetPoint, the <code>oblixOrgPerson</code> type is created. This is an important auxiliary object class. It provides the <code>obUserAccountControl</code> attribute, which you should never modify. This attribute is written to the profile of any user you deactivate.
Group	This type contains information about a group. Examples include <code>groupOfUniqueNames</code> and <code>mailGroups</code> . When you install NetPoint, the <code>oblixGroup</code> and the <code>oblixadvancedgroup</code> type auxiliary classes are created to help you configure useful information on the Group manager.
Location	This type contains information about locations. The Organization Manager uses this object class to store and display location information.
Generic	Any object class that does not fit in the above categories. An example includes the <code>organizationalUnit</code> object class that is managed by the Organization Manager.

Viewing Object Classes

When you install and set up NetPoint, several object classes are already configured. You can view and modify these object classes, and you can create additional object classes.

To view configured object classes

1. From the COREid System Console, click **Common Configuration**.

The **Common Configuration** page appears.

2. Click **Configure Object Class**.

The **Configure Object Classes** page displays object classes that are configured in your LDAP directory and object templates.

▼ Configure Object Class

Configure Workflow Panels

Configure Master Audit Policy

Configure Global Auditing Policies

Configure Object Classes

To change the attribute information, choose an object class and click

Object Class	Object Class Type	Object Class Kind
device	Generic	Structural
document	Generic	Auxiliary
gensiteorgperson	Person	Structural
groupOfUniqueNames	Group	Structural
oblixadvancedgroup	Group	Auxiliary
oblixAuditPolicy	Generic	Don't care Object
oblixAuthenticationPolicy	Generic	Don't care Object
oblixAuxLocation	Generic	Auxiliary
oblixChallengeScheme	Generic	Don't care Object
oblixCustomAuthzCondition	Generic	Don't care Object

This page shows the following information:

Column	Description
Object Class	Name of the object class.
Object Class Type	How the object class is used by NetPoint. See "Object Class Types" on page 64 for details.
Object Class Kind	If you have configured an LDAP object, the kind can be Structural, Auxiliary, or other object. See "Structural and Auxiliary Object Classes in NetPoint" on page 62 for details. An object class kind of Other indicates that the object class kind is undefined. Text can be Don't care or Unknown. If you have configured a template object, the kind can only be Template. See "Template Object Classes" on page 64 for details.
Object Class Attribute	This is used by NetPoint for attribute access and it is also the attribute that NetPoint uses to link search results to a profile page. See "Selecting a Class Attribute" on page 67 for a description.

Modifying Object Classes

From the COREid System Console, you can change the class attribute and the type for an object class. Note that it is important to specify a class attribute for your structural object classes.

You can change the structural object class. However, it is best if you plan your NetPoint configuration so that this task is not necessary.

To modify an object class type

1. From the COREid System Console, click Common Configuration.

The Common Configuration page appears.

2. Click Configure Object Class.

3. Click the link for the object class you want to modify.

The View Object Class page appears.

4. Click Modify.

The Modify Object Class page appears.

5. Select a new class type.

See “Object Class Types” on page 64 for more information on object class types.

6. Save your change.

Selecting a Class Attribute

In the User Manager, Group Manager, and Organization Manager, each tab is associated with a structural object class. Within the structural object class, you select an attribute to be the *class attribute*. The class attribute is used for attribute access. Users who do not have read access to a class attribute have do not have access to the entire entry.

Note: It is not required to set a class attribute for a template object class. You determine user access to template objects and attributes when you configure a workflow, as described in “Chaining COREid Functions Into Workflows” on page 171.

NetPoint also uses the class attribute when displaying search results on a profile page. When a user conducts a search, NetPoint returns a list of results. Each returned item has one value that is displayed as a link. The linked value is taken from the class attribute of the returned object. When the user clicks the link, NetPoint displays the profile associated with that link.

For example, if you specify User Name as the class attribute for the orgPerson object class, when someone searches in the User Manager, the list of search results displays user names as links. Clicking a link displays the profile for that user.

Class attributes are usually selected as follows:

- User Manager uses a class attribute for a person's name.
- Group Manager uses a class attribute for a group's name.
- Organization Manager uses one class attribute per tab. For the location structural object class, the attribute would typically be a location name.

To select the class attribute

1. From the COREid System Console, click Common Configuration.
The Common Configuration page appears.
2. Click Configure Object Class.
3. Click the link for the object class you want to modify.
The View Object Class page appears.
4. Click Modify.
The Modify Object Class page appears.
5. Select the class attribute from the list of attributes.
Note that you can only select a class attribute for a structural object class.
6. Click Save.

Changing the Structural Object Class

Changing the user or group structural object class requires you to rerun COREid System setup to reconfigure NetPoint.

To change user or group structural object classes

1. Shut down all but one COREid Server.
2. Locate *COREid_install_dir/identity/oblix/config/setup.xml*, and change the status parameter value from "done" to "incomplete," as described in "Rerunning NetPoint Setup Manually" on page 299.
3. Restart the COREid Server and navigate to the COREid Administration Console to initiate and complete the setup process as you reconfigure the structural object classes.

When you restart the COREid Server, the other COREid Servers should pick up the new structural user or group object class from the updated Oblix tree.

Adding Object Classes

There are two basic methods for adding an object class in the COREid System Console:

- Configure each attribute manually.
- Select the autoconfigure object class option

This method configures the object class using settings from NetPoint. This option is faster than manual configuration. You cannot view or modify NetPoint-provided attributes before importing them.

With either option, you can later modify the attributes from the System Console. See “About Object Class Attributes” on page 71.

To add an object class

1. From COREid System Console, click Common Configuration > Configure Object Class > Add.

The Add Object Class page appears.

Oblix • NetPoint COREid System Console Logged in
Select Other

System Admin | User Manager Configuration | Group Manager Configuration | Org. Manager Configuration | **Common Configuration**

Configure Object Class

Configure Workflow Panels

Configure Master Audit Policy

Configure Global Auditing Policies

Add Object Class

In the Add Object Class screen, you can configure a new object class. To configure the attributes for the class, select an object class type and click Save. Click Cancel to exit this function without saving the changes.

Schema Domain LDAP

Object Class account

Class Type Person Group Location Generic

Class Kind Structural Auxiliary Template

Auto Configure

The default schema domain is LDAP. If you have not defined any template object classes, LDAP is the only choice. If you have configured additional template object classes and want to configure objects from the additional class, select the class from the Schema Domain list.

2. In the Schema Domain list, select the type of schema that you want to work with, if applicable.
3. From the Object Class list, select the object class to add.

This allows NetPoint to manage the object class. The list contains object classes that were defined in your LDAP directory prior to installing NetPoint.

4. In the Class Type field, select what type of COREid application can manage this object class.
See “Object Class Types” on page 64 for details.
5. In the Class Kind field, select Structural, Auxiliary, Template, or Other.
If NetPoint can determine the Class Kind from the LDAP directory, these radio buttons are hidden.
6. Select Auto Configure object class to populate this object class with attributes from a NetPoint-provided file.
7. Click Save.

When a template object class is saved, it is saved in fully qualified form. For example:

```
obclass=person.miis,o=oblix,o=company,c=us
```

This format is taken from the .tpl file that contains the template object class definition. See “Provisioning External Applications from COREid” on page 253 for details.

How Auxiliary Classes Are Used

You can use auxiliary object classes as mix-ins with structural object classes. This can be helpful when you configure the User Manager, Group Manager, and Organization Manager applications. The more object classes you have at your disposal, the more items you can display on the tabs for these applications, and the more information you can configure for users of those applications.

An object assigned to an auxiliary object class must be associated with a structural class. For example, you can add inetOrgPerson as your structural object class and associate it with the tab in the User Manager application. You can then add auxiliary object classes with attributes for particular kinds of people, such as customers, partners, and so on.

To associate one or more auxiliary object classes with the structural object classes chosen for the COREid applications, see “Adding Auxiliary and Template Object Classes to a Tab” on page 108.

Deleting Object Classes

You can delete auxiliary object classes. You also can delete template object classes that have not yet been added to a tab for a user or group. You cannot delete a *structural* object class. You can only substitute a new structural object class for an existing one. See “Changing the Structural Object Class” on page 68 for details. When you delete an object class, you should also remove any searchbases that you have configured for that object class. See “Setting the Searchbase” on page 128 for details.

To delete an auxiliary object class

1. From the COREid System Console, select Common Configuration > Configure Object Class.
2. Click a link for the object class.
The object class Kind must be Auxiliary.
3. From the View Object Class page, click Delete.

About Object Class Attributes

When installing NetPoint, you configure required structural object classes and their attributes. After completing COREid setup, you may want to add object classes, configure additional attributes, and modify existing attributes. When adding an object class, you can have NetPoint automatically configure attributes in that object class, as described in “Adding Object Classes” on page 69. Use the Modify Attributes feature to change attributes and to configure additional attributes.

The following section describes attribute configuration:

- “About Configuring Attributes” on page 72
- “Attribute Data Types” on page 73

Note: For Active Directory installations, there is a subset of attributes that are unavailable to schema definition by default. To make these attributes visible to NetPoint for configuration, you must remove their entries from the following three files in the *COREid_install_dir/identity/oblix/data/common* directory: *exclude_attrs_config.xml*, *exclude_attrs-ad.xml*, and *ad_exclude_attrs.xml*. Restart the COREid Server for your changes to take effect.

About Configuring Attributes

When configuring an object in NetPoint, you select a class attribute, as described in “Selecting a Class Attribute” on page 67. In addition, you need to decide how NetPoint will display and work with other object attributes. For instance, you need to decide what facts about a person you want the User Manager to display. You also need to decide how you want to display data. For instance, you may want to display lists of printers on the Organization Manager. Or you may want to display a list of preferred travel agents based on a user’s geographical location.

You can configure NetPoint to use any attributes stored in your LDAP directory. Having a well-structured set of attributes to work with allows NetPoint to display the data you want to display and to provide fine-grained access controls for your users.

After configuring an attribute, you must perform additional steps to display the attribute on a profile page in the User, Group, or Organization Manager. For more information, see “Configuring Tab Profile Pages and Panels” on page 113.

After configuring an attribute you must set view and modify privileges to allow users to see the attributes you are displaying. For information about providing users with read and modify privileges, see “Allowing Users to View and Change LDAP Data” on page 126.

Before you configure attributes, you need to understand the relationships between an attribute’s data type, semantic type, and the ability to perform searches. These topics are discussed in the following sections.

Attribute Data Types

When you modify an attribute as described in “About Object Class Attributes” on page 71, a *data type* for that attribute is displayed. A data type is the format of the attribute’s value. For instance, a name attribute may have a data type of a single text line. Every LDAP attribute has an associated data type. In NetPoint, six data types are supported. Data types have corresponding display types, described in “Attribute Display Types” on page 78. You cannot configure the data type for a template or LDAP attribute in the System Console. You configure the data type in the .tpl file or the LDAP schema. Supported data types are shown in Table 7:

Table 7 Supported Data Types

Data Type	Description	Allowed Display Type
String	A case-insensitive or case-sensitive string.	Boolean, check box, date, email address, filter builder, GIF image URL, multi-line text, numeric string, postal address, radio button, selection menu, single line text
Distinguished Name	Distinguished names are how you refer to entries. A distinguished name (DN) is like the path name for a file, except that the DN is read in the opposite order of a path, from the bottom of the directory.	Object Selector, Location (LDAP data only)
Integer	An integer	None, Boolean, check box, date, email address, filter builder, GIF image URL, multi-line text, numeric string, password, postal address, radio button, selection menu, single line text
Telephone	Telephone number	Any display type
Binary	A binary file, such as a GIF file	GIF image, media, password, S/MIME certificate
Postal Address	This is a compound string consisting of one to six sub-strings concatenated with the dollar sign (\$) as the delimiter. Each sub-string can have a maximum of 30 characters.	Postal address

Attribute Semantic Types

A semantic type is an optional characteristic that governs the behavior of the attribute within a COREid application. For example, the value of an attribute assigned to the semantic type Photo appears in the header area of a profile page in a COREid application. You can only assign a semantic type to one attribute. However, an attribute can have more than one semantic type assigned to it. For example, you can assign the Login and DNPrefix semantic types to the cn attribute.

Once a semantic type is assigned, it cannot be assigned to another attribute within a domain unless you disassociate it from the first attribute. For example, only one LDAP attribute can be assigned the semantic type of Password. If you have configured any other schema domain, you could assign the semantic type of Password to only one attribute in that domain. See “Provisioning External Applications from COREid” on page 253 for details.

To disassociate a semantic type from an attribute, you must first specify a semantic type of None for the attribute, and then assign a new semantic type to it.

Each semantic type is associated with one or more display types, as described in “Attribute Display Types” on page 78.

Semantic Types Defined During Setup

Table 8 describes semantic types that are required during NetPoint setup:

Table 8 Semantic Types

Semantic Type	Description	Allowed Display Type
Full Name	Required for the person and group structural object classes and for all structural object classes in Organization Manager. Typically assigned to the cn attribute. The cn attribute is required for most schemas.	Check box, Date, Email address, Multi-line text, Numeric string, Radio button, Selection menu, Single line text
Login	Required for the person object class. Specifies the user credentials during NetPoint login.	Single line text, Email address

Table 8 Semantic Types

Semantic Type	Description	Allowed Display Type
Password	Required for password management for the person object class. It is also required for Active Directory. Specifies the user password for password management. Note: If you are using Sun's iPlanet directory, passwords cannot use UTF-8 characters. If the user supplies UTF-8 characters, the iPlanet directory default 7-bit plug-in fails the operation. By default the 7-bit plug-in requires the uid, mail, and user password attribute values to be 7-bit. To resolve this problem, turn off the plug-in or remove the user password attribute from the configuration.	Password
DN prefix	Required for the person and group structural object classes and for all structural object classes in Organization Manager. Specifies the relative distinguished name (RDN) of an object. The RDN is the leftmost part of the distinguished name (DN). The DN prefix is used when creating an object through a workflow. The attribute with this semantic type must be in the initiating step of a workflow.	Check box, date, email address, multi-line text, numeric string, radio button, selection menu, single line text

Semantic Types Used in Profile Pages

Table 9 shows semantic types used in profile header panels. For more information about profile panels, see “Configuring User, Group, and Organization Manager” on page 99:

Table 9 Semantic Types in a Profile Header Panel

Semantic Type	Description	Allowed Display Type
Photo	Specifies a GIF or JPEG image. The Photo semantic type displays the image in the header of the profile page.	GIF image, GIF image URL
Title	Displays the attribute value in the header of the profile page. Must be associated with a structural class.	Check box, date, email address, multi-line text, numeric string, radio button, selection menu, single line text
Full Name	Is used in a profile header panel and to personalize NetPoint. Users see their name in the NetPoint application user interface.	Check box, date, email address, multi-line text, numeric string, radio button, selection menu, single line text

Semantic Types Used in the Group Manager

Table 10 shows semantic types used in the Group Manager:

Table 10 Semantic Types Used in the Group Manager

Semantic Type	Description	Allowed Display Type
Group Owner	Specifies the attribute where a group owner is stored. NetPoint uses this information primarily as a role in attribute access and delegated administration. Also, group owners can be notified when a user subscribes or unsubscribes from their groups.	Object Selector
Group Dynamic Member	Specifies the attribute storing the dynamic filter that defines the dynamic membership of a group. If you are configuring the Group Manager, you must assign this semantic type to an attribute. The attribute must also belong to the group object class.	Object Selector
Group Static Member	Specifies the attribute where static members of a group are stored. If you are using the Group Manager, you must assign this semantic type to an attribute. The attribute must also belong to the group object class. For NetScape installations, this attribute is uniqueMember. For Active Directory, it is Member.	Object Selector

Location Coordinates Semantic Type

The Location Coordinates semantic type is used to track location. It specifies the position of the location GIF image and is used with the obRectangle attribute. It has no allowable display type because it is used internally by NetPoint.

Semantic Types for Managing Lost Passwords

Two semantic types are used for lost password management. NetPoint provides lost password functionality for both the COREid System and the Access System. Once you configure attributes with these semantic types, end users can enter a challenge-and-response phrase that can later be used to recover their lost passwords:

Table 11 Semantic Types Used for Lost Password Management

Semantic Type	Description	Allowed Display Type
Challenge	Displays a challenge phrase when an end user initiates lost password management. The end user must type the correct response phrase.	Single line text

Table 11 Semantic Types Used for Lost Password Management

Semantic Type	Description	Allowed Display Type
Response	The end user must type the correct response to a challenge phrase when implementing the lost password management feature.	Password

Other Semantic Types

Table 12 describes other semantic types:

Table 12 Other Semantic Types

Semantic Type	Description	Allowed Display Type
Preferred Email address	Used to send email notifications	Email address
Map	This semantic type is used with the location feature in the Organization Manager. When an object is configured to be a location, you should configure a binary attribute to be a map semantic type. The binary attribute stores a GIF or JPEG of a map for the location feature.	GIF image
None	This is a place holder and is not a true semantic type. Select None when you do not want to associate a NetPoint business rule with an attribute.	N.A.

Attribute Display Types

The display type specifies the appearance of an attribute value, for instance, whether the possible values are True or False or an email address. The display type determines whether the attribute can be used when users conduct a search. Only certain display types such as Date and Multi-Line Text are searchable, as indicated in the following table.

Table 13 describes the attribute display types:

Table 13 Object display types

Display Type	Description	Configurable Characteristics
None	A place holder.	N.A.
Boolean	Displays a True or False choice that the user must make. This display type is not searchable.	N.A.
Check Box	Provides a check box. This display type only supports multiple values, and it requires you to specify a rule or a list. See "Using Rules and Lists" on page 83 for details. This display type is not searchable.	Rule (LDAP filter and attribute), List (display name and other features)
Date	Allows users to enter month, day, and year. This display type supports single or multiple values. This display type is searchable.	Data type, data separator
Email	Displays a link to an end user's email address. This display type is searchable.	N.A.
Filter Builder	Creates a button that launches the Filter Builder. The Filter Builder allows users to design custom LDAP queries. This display type is not searchable.	Target object class list
GIF Image	Allows users to find an image. Some NetPoint applications also support JPEGs. This display type is not searchable.	Photo style, height, width
GIF Image URL	Allows you to specify an external location for the GIF image. This allows you to display the image on a profile page. This display type is searchable.	Photo style, height, width
Location	Creates a link in the associated profile page to the Locations page. This display type is used internally in NetPoint.	Target object class list
Media	Used for binary media files. This attribute must have the binary data type. This display type is not searchable.	Description, MIME type

Table 13 Object display types

Display Type	Description	Configurable Characteristics
Multi-Line Text	Two or more lines of text, such as an address. This display type supports single or multiple values. This display type is searchable.	N.A.
Numeric String	Provides a field for specifying a number. This field does not accept non-numeric characters. This display type is searchable.	N.A.
Object Selector	Use the Object Selector display type when you want users to modify an attribute using the Selector. This display type is only valid for attributes of type DN. This display type supports single and multiple values and is not searchable. For more information on the Object Selector display type, see “Search Filters for the Object Selector Display Type” on page 87.	List of object classes, LDAP filter
Password	Lets users type a password. The password characters appear as asterisks, and the user is prompted to enter the password twice. This display type is not searchable.	Text size and length
Postal Address	Six data entry fields in which a user can specify a postal address. Each field can contain a maximum of 30 characters. This display type supports single and multiple values.	N.A.
Radio Buttons	Provides a set of radio buttons that allows the user to select one value from the list of radio buttons. This display type requires you to specify a rule or a list. See “Using Rules and Lists” on page 83 for details. This display type is not searchable.	Rule (LDAP filter, attribute), list (display name, storage name)
Selection Menu	Provides a list. This display type supports single or multiple selectable values. This display type requires you to specify a rule or a list. See “Using Rules and Lists” on page 83 for details. This display type is not searchable. Do not configure DN attributes using the Selection Menu display type. This display type does not support order, which can be important in a DN. For example, if there are two ous in a DN, the order is important.	Rule (LDAP filter, attribute), list (display name, storage name)
Single Line Text	Displays information in a single line of text. There is no maximum number of characters for this field. This display type supports either single or multiple values. This display type is searchable.	N.A.

Table 13 Object display types

Display Type	Description	Configurable Characteristics
S/MIME Certificate	Stores certificate data in the configured attribute rather than on disk. This display type is not searchable.	N.A.

Note: Once you have assigned an attribute to a COREid application panel as described in “Configuring Tab Profile Pages and Panels” on page 113, if you want to change the attribute display type or semantic type you must delete the panel, change the attribute display type, and re-create the panel.

Viewing Attributes

You view attributes on the Modify Attribute page.

To view the Modify Attribute page from the System Console

1. After logging in to the COREid System Console, click Common Configuration > Configure Object Class.

The Configure Object Classes page appears.

2. Click an object class.

The View Object Class page for the selected class appears.

3. Click Modify Attributes.

The Modify Attributes page appears.

To view an application-specific Modify Attribute page

1. Navigate to the COREid System Console.
2. In the System Console, click the User Manager Configuration tab (or Group Manager Configuration or Organization Manager Configuration).
3. In the side navigation bar, click Configure Tab.

The Configure Tab page appears. The structural object class for that tab is displayed as a link under the heading “Existing Tabs.”

4. Click the link under the Existing Tabs label.

The View Tab page appears.

5. Click the Modify Attributes button.

The Modify Attributes page appears, as shown next.

The screenshot shows the 'Modify Attributes' page in the COREid System Console. The page title is 'Modify Attributes'. Below the title, there is a brief description: 'Through Modify Attributes, you to modify display name, semantic type, display type, and attribute value(s) for the attributes Employees tab. After modification, please click done button to save the attribute information.'

The form contains the following fields:

- Attribute:** A dropdown menu with 'audio' selected. The list of options includes: audio, businessCategory, carLicense, cn, departmentNumber, description, and destinationIndicator.
- Display Name:** A text input field containing 'Audio'.
- Semantic Type:** A dropdown menu with '(None)' selected. Other options are Map, Photo, and Response.
- Data Type:** A dropdown menu with 'Binary' selected.
- Display Type:** A dropdown menu with 'Media' selected.
- Description:** A text input field containing 'Music File'.
- Mime Type:** A dropdown menu with '(None)' selected.
- Attribute Value(s):** Radio buttons for 'Single' (selected) and 'Multiple'.

Configuring Attributes

When installing NetPoint, you configure all required attributes for your structural object classes. After installation, you can modify attributes to resolve conflicts among configured attributes and to configure additional attributes.

Note: For Active Directory installations, there is a subset of attributes that are, by default, unavailable to schema definition. To make these attributes visible to NetPoint for configuration, you must remove their entries from the following three files in the *COREid_install_dir/identity/oblix/data/common* directory: *exclude_attr_config.xml*, *exclude_attr-ad.xml*, and *ad_exclude_attr.xml*. You must restart the COREid Server for your changes to take effect.

To configure an attribute

1. Open the Modify Attributes page as described in “Viewing Attributes” on page 80.
2. In the Attribute list, select an attribute you want to modify.

The attribute’s data type appears below the list. This is a read-only field.

Note: Novell Directory Server (NDS) maps the attribute and object class names from the native directory server to the LDAP layer of NDS. Some of these attributes or object classes will have multiple mappings (aliases) in the LDAP layer. For example, the native NDS object class is `group`, while the LDAP layer of NDS maps two aliases called `GroupofNames` and `GroupofUniqueNames`. For NetPoint to work correctly, make sure the object class or attribute name that you provide during configuration is the one that occurs ahead of the other mappings for the same object class or attribute. You can check the mapping order through `consoleOne`.

3. In the Display Name field, enter a user-friendly display name for this attribute.

The display name appears on a COREid application page, for instance, the User Manager. For example, for the `departmentNumber` attribute, you might enter Department Number as the display name.

For template object attributes, the Display Name should indicate the template being used. As noted earlier, users will not be able to see the data values for these attributes. As a result, you should identify these “special case” fields so that users can be advised that it is normal for data not to be shown. For example, in an object template for MIIS, you might want all MIIS-related attributes to be identifiable by their display names, such as `assistant.person.miis`.

4. The Data Type field displays the attribute’s data type, as described in “Attribute Data Types” on page 73.

This is a read-only field.

You cannot use attributes with binary, distinguished name, or postal address data types as report criteria or in search attributes.

5. In the Semantic Type list, you can optionally select one or more semantic types.

See “Attribute Semantic Types” on page 74 for details on semantic types.

6. In the Attribute Values field, specify whether the attribute can have single or multiple values.

Depending on the attribute, data type, and display type, this option may not be available.

7. In the Display Type list, select the attribute's display type.

If you select a date attribute for the display type, you must select a date type to indicate how you want the date to appear on the COREid application profile page. Do not change the date type after you select it because this may display existing data incorrectly.

Several display types allow you to specify a rule or list. See “Using Rules and Lists” on page 83 for details.

Other display types allow you to specify a photo or text. For more information on these fields, see “Configuring Other Display Types” on page 93.

Using Rules and Lists

The display types of Selection Menu, Radio Buttons, and Check Box require that you specify a *rule* or a *list*. For instance, you may assign a data type of string and a display type of radio button to a Title attribute. To display a list of titles on a User Manager profile page, you need to build a list.

A *list* is a static set of values. A *rule* is an LDAP filter that queries the directory before building a list. For example, if you create a filter to find every instance of `objectClass=dialUpConnection` with an attribute of `TelephoneNumber`, a list of phone numbers is shown in the selection menu.

For more information about LDAP filters, refer to “Search Filters for the Object Selector Display Type” on page 87 for information. Also, the Internet Engineering Task Force's RFC 2254 defines a string representation of LDAP search filters.

Defining a Rule

You can define a static list to display on a NetPoint application page, or you can define a rule. For instance, you can provide a static list of available printers, or you can construct this list from entries for printers in your directory. When you configure a rule, you create a dynamic list based on entries in your directory. A rule is a directory query based on an attribute that you specify in the rule. The query returns a set of records from the directory. Using the rule, you cause NetPoint to build the list to be displayed on the application page by doing the following:

- Querying the directory for a specific object or attribute
- Building a list of hits
- Selecting one attribute from each directory hit
- Building a list showing the values for each attribute

The advantage of a rule over a list is that what is displayed as a result of the rule filter is updated whenever the directory is updated.

To define a rule

1. Navigate to the Modify attributes page.
2. Select an Attribute from the list.
3. From the Display Type list, select the attribute's display type.

A Rule button, Add Filter text box, and Attribute field appear, as shown below.

The screenshot shows a web interface for defining a rule. At the top, there is a 'Display Type' dropdown menu set to 'Radio Buttons'. Below this, there are two radio buttons: 'Rule' (which is selected) and 'List'. To the right of the 'Rule' radio button is a large empty text box labeled 'Add Filter'. Below the 'Add Filter' box are three input fields: 'Attribute', 'Display', and 'Storage'. The 'Attribute' field is currently empty. Below these fields is another large empty text box.

4. Select the Rule button.
5. In the Add Filter box, type an LDAP filter.

For example:

(objectclass=printer)

Suppose you invoked the Modify Attribute page for an attribute called Printer, with a display name of Printers. The rule shown above would be appropriate for displaying a list of printers.

For examples of filters, see “Search Filters for the Object Selector Display Type” on page 87.

6. In the Attribute field, type the LDAP name of the attribute that you want to associate with the filter.

In the printer example, you might type PrinterName. This rule causes an LDAP query on the printer object class and builds a list of values taken from the PrinterName attribute of each returned entry.

7. Continue with “Defining a List” on page 85.

Defining a List

A list is a static set of values presented to a user.

To define a list

1. On the Modify attributes page, click the List button.

List-related Display and Storage fields become active.

The screenshot shows a configuration window with a 'Display Type' dropdown set to 'Radio Buttons'. On the left, there are two radio buttons: 'Rule' (unselected) and 'List' (selected). Under 'Rule', there is an 'Add Filter' button and a large empty text area. Under 'List', there are three input fields: 'Attribute', 'Display', and 'Storage'. Below these fields are four buttons: 'Add', 'Move Up', 'Move Down', and 'Delete'. At the bottom of the window are three buttons: 'Save', 'Done', and 'Reset'.

2. In the Display field, type the list item's Display Name.

This is a name that the user sees when this attribute is displayed on an application page, for instance, the User Manager.

3. In the Storage field, type a storage name for the attribute.

This value is saved in the database. It can be the same as the display name, or it can follow your own database-naming conventions.

When you click Add, if you omit a storage name, the display name is used as the storage name.

If you want to change a storage name, delete the entry in the Storage field, and retype the Display and Storage names.

4. Click Add.

The information is added to the List field.

Items in the list appear on the NetPoint application page in the order they appear on this page. To rearrange items in the list, or to remove an item, use the Move Up, Move Down, and Delete buttons.

Localizing Attribute Display Names

You can localize attribute display names to present to COREid applications to end-users in their native language. See “Configuring Multiple Languages for NetPoint” on page 272 for information on managing multiple languages.

In order to localize object class attributes, you must manually enter the attribute display names in the COREid Console for each language that you installed. After you have localized object class attributes, you can view and modify them in the COREid System Console.

The process for localization is the same for LDAP and template objects.

Table 14 lists the attributes that can be localized for each object class.

Table 14 Items You Can Localize

Items	What You Can Translate
Object classes configured for tabs	Name Description Mouseover
Object classes configured for panels	Name Description Mouseover
Attributes	Display name
Attributes with a media display type	Display name
Attributes with a choice display type	Display name
Workflow definitions	Workflow name Note: You specify a translation for your workflow name when you create or modify a workflow definition, as described in “Chaining COREid Functions Into Workflows” on page 171.

To create, view, or modify localized attribute display names

1. From the COREid System Console, click Common Configuration.
The Common Configuration page appears.
2. Click Configure Object Class.

3. Click the link for the object class you want to modify.

The View Object Class page appears.

4. Click Translate.

The Summary of Attribute Display Names page appears. All language-specific attribute display names are displayed on this page. Display names that have not been configured are marked as Not Configured.

5. Click Modify to create or modify a display name.

The Configure Attribute Display Names page appears. This page lists links for installed languages and the localized display names for attributes.

6. Click the language for which you want to modify attribute display names.

7. Enter display names in the Display Name fields.

8. Click Save to save your changes (or Cancel to exit the page without saving your changes).

Note: If a display name has not been configured for a language, a localized “Not Configured” message is displayed in the display name field.

Search Filters for the Object Selector Display Type

The object selector display type associates a Selector with the LDAP attribute assigned to this display type. (This does not apply to template attributes.) Users invoke the Selector to search for users or groups. The Selector is available when you view, create, or modify a profile or a workflow. Selector buttons are shown below:



Use the object selector display type to create a search filter for the Selector. See “The Selector” on page 40 for details. You can write search filters to help people conduct a Selector search during the following operations:

- Create profile
- Modify profile
- Modify workflow
- Delete workflow

These filters do not apply to the Query Builder.

When a user invokes the Selector, they perform a directory search. When you create a filter for the Selector, the filter is used in an “and” relationship with the search criteria that the user provides.

A filter can be static. For example, you can restrict Selector searches so that the search results only contain people with an organizational unit of Corporate in their directory profile.

A filter can also be dynamic. For example, you can restrict a Selector search to return only people whose organizational unit matches that of the person being displayed on the Modify Profile page where the search was invoked. When using a dynamic filter for a Modify Profile page, the Selector search is based on the directory profile of the person being displayed. In the case of creating profiles, a dynamic filter produces Selector search results based on the login profile of the person performing the task.

Creating a Search Filter for the Object Selector Display Type

A filter helps the user narrow down a search. A filter narrows down the place in the directory tree where a search may be conducted.

To create a filter

1. From the COREid System Console, click Common Configuration > Configure Object Class > *Link* > Modify Attribute

For example, to develop a search filter for a sales person, you might navigate to the Modify Attribute page for the Person object class.

2. On the Modify Attribute page, in the Attribute list, choose the attribute for which you want to define a Selector search.

You must choose a DN attribute. For instance, if you want Selector searches for sales people, you might select a DN attribute called salesPersonDN.

3. On the Display Type list, select Object Selector.

If the attribute you chose in the previous step is a DN attribute, the Object Selector option appears in the list. The Target Object Class list and the Add Filter input box are also displayed.

4. In the Target Object Class list, select an object class to be used as the primary key in the search filter.

The target object class determines what is displayed on the Selector search page. For instance, if you want the Selector to help users find sales people, you might use Person as the target object class. If you select more than one Target Object Class on the Modify Attribute page, the Selector application will contain a tab for each object class.

5. Type a valid LDAP filter in the Add Filter input box.

The filter determines what is displayed on the Selector search results. For examples of the types of LDAP filters you can write, see “Static LDAP Search Filters” on page 90 and “Examples of Dynamic LDAP Search Filters” on page 92.

Note that your filter can use only attributes that are contained in the definition of the target object class.

Note: NetPoint treats white spaces literally. Be aware of extra trailing spaces or carriage returns in your filters.

6. Save your changes.

This creates a filter that is used in an *And* relationship with any other criteria the user specifies on the Selector search.

Search Filters for Multiple Target Object Classes

If you select more than one target object class on the Modify Attribute page, the Selector application will contain a tab for each object class. Your search filter will need to contain an *Or* operator (|) and separate selection criteria for each object class you selected. An example of this type of search filter is provided in “Static Searches Using Multiple Target Object Classes” on page 91.

Deleting a Search Filter

Remove a filter by erasing the text in the Filter text box.

Usage of Rules and Filters

This section covers important topics related to rules and filters:

- Creating basic static filters
- Creating dynamic filters using substitution syntax
- Use of wildcards in a search
- Proper use of the Not operator when writing a filter

Static LDAP Search Filters

When you implement a static search filter, all search results must match a fixed value. For example, you can restrict a search to return only people whose directory profiles show an organizational unit of Sales.

As an example of a simple static filter, suppose you want to provide Selector searches for the `seeAlso` attribute. The filter will return search results that show only people whose directory profiles contain a `businessCategory` value of *dealership*.

To create a static filter

1. Navigate to the Modify Attribute page for the object type that triggers this search filter (for instance, Person).
2. Select the “seeAlso” attribute in the Attribute list.
3. From the Display Type list, select Object Selector.
4. In the Target Object Class, select the object class that will be the primary key for the filter (for instance, Person).
5. In the Filter text box, input the following:
(businessCategory=dealership)

Static Searches Using Wild Cards

As an example of a static filter that uses wild cards, suppose you want only people with the word *Manager* in their title to be returned on a search using the Selector. You can create a filter that searches for the string Manager.

To create a static search filter using a wild card

1. Navigate to the Modify Attribute page for the object class containing the DN attribute to be associated with the Selector (for instance, the Person object class).
2. In the Attribute list, select the DN attribute (for instance, the Manager attribute).
3. On the Display Type list, select Object Selector.
4. In the Filter text box, input the following:
(attribute=*value*)
For example:
(title=*manager*)

Static Searches Using Multiple Target Object Classes

You can build a static filter that searches for more than one target object class. For example, suppose you want to build a filter for the uniqueMember attribute so that a search using the Selector returns one of two items:

- On a search of people, the search results show only full time employees
- On a search of groups, the search results show only mail groups

In this example, to create an LDAP filter for both characteristics, you need to select people with employee=fulltime in their directory profiles, and you need to select groups with the object class of MailGroup in their directory profile. Each attribute in the filter must be associated with an appropriate object class. Finally, you need to join the two searches with an Or operator (|), as follows:

```
( | (&(objectclass=inetOrgPerson)(employeeType=FullTime))
  (&(objectclass=groupOfUniqueNames)(objectclass=MailGroup)))
```

Substitution Syntax

You can enter *substitution syntax* using the Advanced button of the Query Builder. See “Writing LDAP Filters Using Query Builder” on page 132 for details.

When using substitution syntax, the variable attribute value for the source DN (the person logged into the application) is substituted in the rule and evaluated against the target DN (the entry you are trying to view or modify).

Substitution syntax allows a rule to be evaluated dynamically, according to the person executing a task. The syntax is as follows:

```
attribute=$attribute$
```

For example:

```
(ou=$ou$)
```

This rule filters all those in the same organizational unit as the person logged into the application.

You can use operators such as *And* and *Or* in a filter. For example:

```
( | (ou=$ou$) (ou=people))
```

Note: For the selected searchbase, users can search only for entries from the same ou as their own. Additionally, users from ou=people can search for entries within the selected searchbase.

Examples of Dynamic LDAP Search Filters

In addition to specifying a conventional LDAP search filter, you can use NetPoint's filter substitution syntax to create a dynamic filter. A dynamic filter allows a search to return results that are based on a user profile. For instance, suppose you create a search filter for the `orgPerson` attribute that contains the following:

```
(ou=$ou$)
```

Using this search filter, if you conduct a Selector search on a Modify Profile page for a person, your search results contain only people whose directory profiles match the organizational unit in the profile of the person you are modifying. For example, if you invoke the Modify Profile page for John Smith and invoke the Selector to choose John Smith's manager, the search results show only people in John Smith's organizational unit.

When you use filter substitution, the profile of the search target is substituted. In the example above, the value of John Smith's `ou` is substituted. If a target is not present, for instance, in the Create Workflow function, the search filter substitutes the value from the login profile of the person creating the workflow.

For example, suppose you are creating a workflow for a group named Corporate whose organizational unit is not yet defined in the directory server. In this case, NetPoint uses the `ou` value of the logged-in participant who is creating the group. The logged-in participant's `ou` value carries over in the workflow until you commit this group in the directory server. At that point, the `ou` value in the dynamic filter (`ou=ou`) changes to the group's `ou` value.

As another example, suppose you want people who have the Secretary attribute in their directory profile to receive search results containing only people who have the same manager that they do. From the Configure Attribute page for the Secretary attribute, you would specify the following:

```
(manager=$manager$)
```

Dynamic Searches Using Wild Cards

You can use wild cards in a dynamic filter. For example, suppose you want to supply a `contactPerson` attribute in an `organizationalUnit` object. The `contactPerson` attribute should return people in same Zip code as the `organizationalUnit` object. If the `organizationalUnit` profile contains an attribute called `zipCode`, and the Zip code is specified at the end of a `postalAddress` directory attribute, you would specify the following in the filter:

```
(postal Address=*$zipCode$)
```

Dynamic Searches Using Multiple Values

Finally, you can supply multiple values in a dynamic search filter. For example, suppose you want the `seeAlso` attribute for business objects to select the `businessCategory` of dealership, and specifically, dealerships in the same state as the search target. You would specify the following filter for the `seeAlso` attribute:

```
(&(businessCategory=dealership)(state=$state$))
```

Use of the Not Operator

You use And (&) and Not (!) operators when constructing a filter. For example:

- **Example of an And Operation**—`(&(sn=white)(objectclass=personOC))`
- **Example of a Not Operation**—`(&(!(sn=white))(objectclass=personOC))`

In the example of the Not filter, above, you might expect the following filter to be valid:

```
(!(sn=white))
```

However, when specifying a Not operation, you need to embed it within an And and specify the person object class. A filter such as `!(sn=white)` is not allowed because a search of this type would be conducted on the entire domain before targeting the reduced domain set specified on the filter. This is costly steps in terms of performance. The optimized algorithm that NetPoint uses causes the search to be conducted on the reduced domain set. The proper use of the Not operation is as illustrated below:

```
(&(!(sn=white))(objectclass=personOC))
```

The optimized algorithm causes the filter `!(sn=white)` to not give the expected result.

Configuring Other Display Types

There are configuration options for the other display types.

To configure a GIF image display type

1. On the Modify Attributes page, select the Attribute Photo.
2. Using the buttons under the Display Type list, select the photo style:
 - **True Size**—Select True Size to display the GIF in its actual size.
 - **Fixed Size**—Select Fixed Size to specify the height and width for the image.

When you select a text-based display type, you can use XSL style sheets to configure the defaults for the text. This is also true for setting columns and rows. See the *NetPoint 7.0 Customization Guide* for details.

When you select an attribute with a display type that uses target object classes, you specify one or more objects for association with this attribute. This display type supports single or multiple values.

- If you want to set a target object class, select one or more required object classes in the Target Object Class list.
- If you want to set the MIME type, select the kind of media file you want to attach from the MIME Type list.

Configuring Derived Attributes

Derived attributes are virtual LDAP object class attributes whose values are generated by comparing the contents of one object class's attribute against an attribute in the same or a different object class. The main purpose of a derived attribute is reverse lookup. For example, the profile of a person may contain the name of their administrative assistant, but administrative assistants rarely have the names of all of the people they administer in their profiles. A derived attribute allows the administrative assistant to refer back to all of the people who have the administrative assistant in their profile. Similarly, a `groupOfUniqueNames` object may contain a `uniqueMember` attribute with links to members of the group. But a person object might not link back to the `groupOfUniqueNames`. The derived attribute feature would allow group members to refer back to the group they belong to.

Creating derived attributes involves specifying two attributes whose values are to be compared. All matches are added to the derived attribute.

Use derived attributes to provide information in profiles that otherwise would require an LDAP filter, search, or report.

Attributes associated with template objects cannot be configured as derived attributes.

Note: Using derived attributes can result in slower response times, especially if you have multiple attributes or if your attribute references multiple object classes (such as a Group Manager derived attribute performing a lookup on a User Manager attribute). If you experience performance issues with your derived attributes, Oblix recommends you modify your index file to include the corresponding attribute index and reimport it.

Example of a Derived Attribute

Suppose the administrative assistants of your organization have asked to view all the managers they support in the User Profiles in User Manager. To do this, you can create a derived attribute that takes the attribute of each administrative assistant and compares it with the value for Secretary in everyone else's profile. When an administrative assistant views user identities in User Manager, only people who match the derived attribute are displayed.

Here is a summary of the steps required to create a derived attribute.

To create a derived attribute

1. Give your new attribute a descriptive display name, such as My Direct Reports.
2. Specify Self as the Match Attribute.

This indicates that the administrative assistants' DNs are the search criteria.

3. Because you are looking for your administrative assistant's direct reports, specify Person object class as the searched object class.

Be sure you are looking for people and not groups or other kinds of objects.

4. Specify secretary as the Lookup Attribute.

You are searching the user identities for users with matching values in the secretary attribute.

5. Save your new attribute and associate it with the Employee tab in User Manager.

Now, whenever an administrative assistant views the User Profile page, NetPoint takes the value of the Self attribute (DN) and compares it against the values of everyone's secretary attribute. Wherever there is a match, NetPoint lists that manager's name in the administrative assistant's My Direct Reports section of the User Profile page.

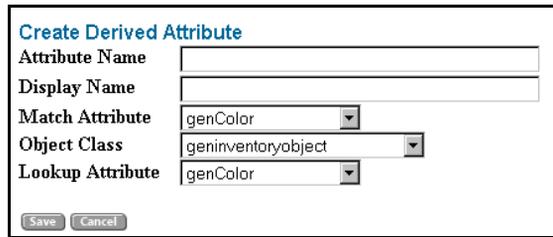
Note: The attributes displayed in the profile are determined by the selected object class's Object Class Attribute. To change this value, you must modify the object class.

To configure a derived attribute

1. From the COREid System Console, click Common Configuration.
The Common Configuration page appears.
2. Click Configure Object Class and click the named link of an object class.
The View Object Class page appears.

3. Click Modify Derived Attributes > Add.

The Create Derived Attribute page appears.



Create Derived Attribute

Attribute Name

Display Name

Match Attribute

Object Class

Lookup Attribute

4. In the Attribute Name field, specify a name for your new derived attribute.

Note: Since a derived attribute is a virtual attribute, the attribute name must not exist in the schema.

5. In the Display Name field, type the name of the derived attribute as it will appear in NetPoint pages.
6. In the Match Attribute field, select the attribute in the current object class whose values you want to match.
7. In the Object Class field, select the object class you want to search.
8. In the Lookup Attribute field, select an attribute in the specified object class whose values you want to compare against.
9. Click Save to save your changes (or Cancel to exit without saving).

Assigning a Derived Attribute to a User Manager Tab

Before you can use your derived attribute, you must assign it to a COREid application tab.

To add a derived attribute to an application tab

1. From the COREid System Console, click User Manager Configuration (or Group Manager Configuration or Organization Manager Configuration).
2. Click Configure Tab.
3. Click the tab link, then click View Object Profile.

The page that appears depends on which application you are using. In User Manager, the Configure User Profile page appears. In Group Manager, the Configure Group Profile page appears. In Organization Manager, the Configure Object page appears.

4. Click the tab for configuring panels.

In User Manager and Organization Manager, it is the Configure Panels tab. In Group Manager, it is Configure Group Profiles Panel.

The Configure Panels (or Configure Group Profile Panels) page displays the panels currently configured to appear in a User Manager profile.

5. Click the panel you want to add the derived attribute to.

The View Panel page appears.

6. Click Modify.

The Modify Panel page appears.

7. In the Attributes menu, select the derived attribute you want to add.

8. In the associated text box, type the name as you want it to appear in NetPoint pages.

9. If you need additional Attribute fields, click Add.

10. Click Save to save your changes (or Cancel to exit without saving).

Permissions for Derived Attributes

You can assign Read permissions to a derived attribute. See “Setting and Modifying LDAP Attribute Permissions” on page 138 for information about assigning rights to an attribute.

Attributes Configured on a Per-Application Basis

When you configure attributes as described in this chapter, these attributes are available to any COREid application. In other words, the attribute can be used within the User Manager, Group Manager, or Organization Manager.

However, you may want to make only certain attributes available to certain applications. For example, you may want an attribute for a person’s address to only be available to the User Manager application. If this is the case, you can access the object and attribute configuration functionality described in the preceding sections from the relevant application. In the case of the Organization Manager, you can configure objects and attributes on a per-tab basis.

For more information on configuring the User, Group, and Organization Managers, see “Configuring User, Group, and Organization Manager” on page 99.

4 Configuring User, Group, and Organization Manager

The User Manager, Group Manager, and Organization Manager are COREid applications that enable end users to view and modify information about themselves, other people, groups, inventory, and any other item that you, the administrator, choose to make available.

The chapter on “Making Schema Data Available to NetPoint” on page 59 describes how to make NetPoint aware of objects and attributes in your directory and in object template files, and how to configure the way that attributes are displayed on an application page. This chapter explains how to place attributes on specific application pages, and how to enable users to view and modify them. This chapter also touches on end use of these applications.

You must be a NetPoint Administrator or Delegated Identity Administrator to configure the User Manager, Group Manager, and Organization Manager applications. See “Delegating Administration” on page 47 for details.

This chapter covers the following topics:

- “About User, Group, and Organization Manager” on page 100
- “Configuring Tabs” on page 101
- “Configuring Tab Profile Pages and Panels” on page 113
- “Allowing Users to View and Change LDAP Data” on page 126
- “Examples of Configuring an Application” on page 143
- “End-User Scenarios” on page 147
- “Configuring Logging and Auditing Policies” on page 153
- “Generating Reports” on page 156
- “Advanced Configuration” on page 161

About User, Group, and Organization Manager

The User, Group, and Organization Manager are the primary COREid applications:

- People use the User Manager to view information about their identity, to modify information such as their home phone number, and to find information on other people.
- People use the Group Manager to view groups, subscribe to groups, and to manage group subscriptions.
- People use the Organization Manager to manage other objects.

Popular uses of the Organization Manager include viewing organization maps and searching for inventory items.

You control who is allowed to see what attributes and values on these applications. You also control what portion of the directory tree is accessed when users conduct searches. You can add filters so that when users search, the results conform to criteria specified on the filters.

When you first install, set up, and configure objects and attributes in the COREid System, the COREid application pages are empty. You make information available on a COREid application as follows.

Task overview: Displaying information on an application

1. Configure the objects and attributes to be used by the COREid applications, as described in “Making Schema Data Available to NetPoint” on page 59.
2. Configure the COREid application pages, or *tabs*, as described in “Configuring Tabs” on page 101.
3. Configure profile pages on each tab by arranging groups of attributes into panels, as described in “Configuring Tab Profile Pages and Panels” on page 113.
4. Optionally, set the searchbase to control what portion of the directory tree is included in a search for LDAP attributes only, as described in “About the Searchbase” on page 126 for details.
5. Set permissions for users to view and modify the attributes you are displaying on the application tabs, as described in “About View and Modify Permissions” on page 137 for details.

Configuring Tabs

The COREid applications each have one or more tabs, which are configured as follows:

- When you work with the Group Manager Configuration tab in the COREid System Console, you are configuring the My Groups tab in the Group Manager application.
- Similarly, the User Manager Configuration tab in the COREid System Console is used to configure the My Identity tab in the User Manager application.
- Unlike the User Manager and the Group Manager, you can configure multiple tabs in the Organization Manager.

When you install NetPoint, a default structural object class Location is defined for the Organization Manager. The Organization Manager can manage objects defined in COREid as having a Generic or Location data type. See “Object Class Types” on page 64 for details.

The tab in the User Manager is associated with the person structural object class. The tab in the Group Manager is associated with the group structural object class. The Organization Manager can have multiple tabs, each associated with a different object class. All tabs may have auxiliary LDAP object classes and template object classes associated with them.

Viewing and Modifying Tab Configuration Information

You can view and modify characteristics of the tabs that are displayed on the User, Group, and Organization Manager pages.

To view or modify tab configuration information

1. Navigate to the COREid System Console and click User, Group, or Org. Manager Configuration.

The User, Group, or Organization Manager Configuration page appears.

2. Click Configure Tab.

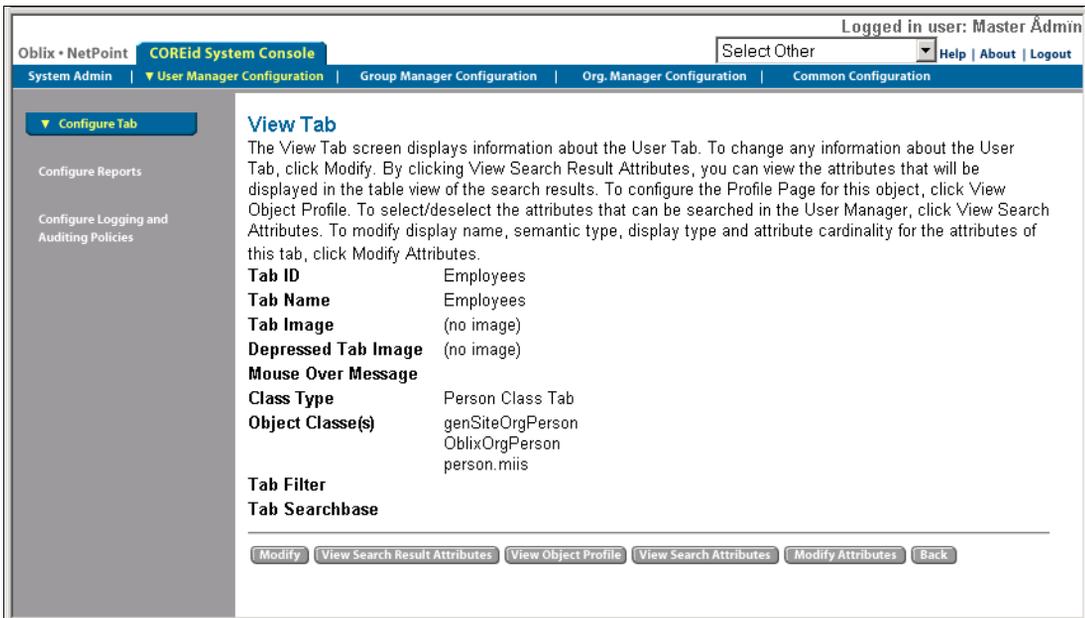
The Configure Tab page appears, showing the name of the tab for the application. The Organization Manager may have more than one tab.



3. Click the link to the tab.

Since there is only one tab for User Manager and Group Manager, there can only be one link. For Organization Manager, there can be more than one tab.

The View Tab page appears.



This page contains the following information:

Field	Description
Tab ID	Unique identifier for the tab.
Tab Name	The name displayed on the application tab. You can localize this field.

Field	Description
Tab Image	<p>GIF image for the tab. The GIF must be stored in <i>WebPass_install_dir/identity/oblix/lang/langTag/style0</i> where <i>WebPass_install_dir</i> is the directory where you installed WebPass and <i>langTag</i> is the folder that contains the specific language that you are using.</p> <p>Enter only the name of the GIF file, not the full path.</p>
Depressed Tab Image	GIF image displayed when a user clicks the tab image.
Mouse Over Message	<p>Text displayed when the user passes the cursor over the tab.</p> <p>You can localize this field.</p>
Class Type	The type associated with the structural class for this tab. See “Object Class Types” on page 64 for details.
Object Class(es)	<p>The structural, auxiliary, and template object classes used by this tab. Template object classes are shown in fully qualified format, for example, <i>miis.person</i>. The format is read from the <i>.tpl</i> file where the class is defined. See “Provisioning External Applications from COREid” on page 253 for details.</p> <p>You cannot change the structural object class through COREid System Console. You can associate auxiliary object classes with the structural object class, as described in “Adding Auxiliary and Template Object Classes to a Tab” on page 108.</p> <p>Note that some object classes may appear in a non-editable list on this page, and other object classes may appear in a text box on this page. The object classes in the text box have not yet been added to the tab.</p>
Tab Filter	<p>An LDAP filter that queries the directory and returns objects qualified by the filter. For examples of the types of LDAP filters you can write, see “Static LDAP Search Filters” on page 90 and “Examples of Dynamic LDAP Search Filters” on page 92.</p> <p>Tab filters do not support filter substitution. Tab filters affect searches, viewing and modifying profiles, and creating reports on the tab. The filter is used in an “and” relationship (with criteria specified during a search) and when creating reports. That is, the criteria from both the filter and the search are applied. View and modify operations use this filter to qualify the target object.</p>
Tab Searchbase	Starting point in the directory tree (DIT) for user searches. See “About the Searchbase” on page 126 for details.

4. Click Modify
5. Make the desired changes and click Save.

If you do not see your changes reflected in the COREid application, go to COREid System Console > System Configuration > View Server Settings > Clear Cache to flush and reload the cache.

Note: When you modify a tab image, depressed tab image, and so on, these elements are immediately available for users to view. This is different from adding attributes to a panel, which requires setting permissions before users can view the information.

Localizing Tabs

If you have installed more than one language pack, you can localize tab names to display them in those languages. You create, view, and modify localized tab names in the Administration Console.

See “Configuring Multiple Languages for NetPoint” on page 272 for information on managing multiple languages.

To create, view, and modify localized tab configuration

1. Log in to the COREid System Console and click User, Group, or Org. Manager Configuration.

The User, Group, or Organization Manager Configuration page appears.

2. Click Configure Tab.

The Configure Tab page appears, showing the name of the tab or tabs for the application.

3. Click an existing tab to view its details.

The View Tab page appears. Tab details such as the ID, name, class type, and object classes are displayed on this page.

4. Click Translate.

Note: The Translate button appears only if more than one language has been installed.

The Summary of Tab Label Display Names page appears. Display names, if any, that have been configured for the following language-specific fields appear on the page:

- Tab Name
- Mouse Over Message

Display names that have not been configured for a particular language are marked as Not Configured.

5. Click Modify to create a tab display name or to modify an existing one.

The Configure Tab Display Names page appears. This page contains fields for the tab display names and links for all the installed languages.

6. Click the language for which you want to localize the tab.
7. Enter the display names in the Tab Name and Mouse Over Message fields.
8. Click Save to save your changes (or Cancel to exit the page without saving your changes).

Adding a Tab to the Organization Manager

The Organization Manager can contain more than one tab.

To add a tab

1. From the COREid System Console, click Org. Manager Configuration > Configure Tabs.

The Configure Tabs page appears.

2. Click Add.

The Create Tab page appears.

3. Complete the fields in this page, as described in “Viewing and Modifying Tab Configuration Information” on page 101.
4. Click Save.

Specifying the Search Attributes on a Tab

At the top of the application page for the User Manager, Group Manager, and Organization Manager there are search fields. See “Adding Special-Purpose Object Classes to a Group Tab” on page 109 for an example. You specify what attributes appear in the search function drop-down list. Note that search attributes can only be taken from an LDAP directory. Template attributes cannot be used as search attributes.

Note: You must configure attributes before they can appear on a tab. For more information, see “About Object Class Attributes” on page 71.

To specify what attribute can be used in a search

1. From the COREid System Console, click User, Group, or Org. Manager Configuration > Configure Tab > click a tab link > View Search Attributes.

The View Search Attributes page appears.

2. Click the Modify button at the bottom of the page.
3. Select an attribute check box to make the attribute searchable.
4. Click Save.

Viewing, Modifying, and Localizing Search Result Attributes

You choose what attributes are to appear in the results of a search. If you have installed and configured multiple languages, you can localize search result attributes. This enables you to display search results in multiple languages.

To view the Search Result attributes

1. From the COREid System Console, select User, Group, or Organization Manager Configuration > Configure Tab > *TabName*.

The View Tab page appears.

2. Click View Search Result Attributes.

The View Search Result Attributes page appears.

This page shows the attributes that appear when the results of a user’s search are displayed. If you have configured COREid for more than one language, those languages are displayed on the page.

3. Click Modify to change the attributes.

The Modify Search Result Attributes page appears.

Modify Search Results Attributes
 Choose the attributes to be displayed in the search results..
 The first attribute is always the Class Attribute defined for the Object Class. You can choose 4 more attributes.

Attributes	Name	
	Name	Name
Description	Description	Description
Admin	Admin	Admin
Business Category	Business Category	Business Category
Building Number	Building Number	Building Number

Save Cancel Add

4. The first attribute is always the Class Attribute.

You cannot modify the class attribute **Name** from this page. It is displayed in bold and is not editable on this page. If you want to modify a class attribute, see “Selecting a Class Attribute” on page 67 for details.

5. From the lists on the left, select new attributes for each field you want to change.

The attribute’s Display Name appears in the corresponding field on the right. This name appears in the COREid user application. See “About Object Class Attributes” on page 71 for details.

6. Click Add if you need additional attribute fields.
7. Click Save.

To localize search result attributes

1. In User, Group, or Organization Manager > Configure Tab > click a link.
The View Tab page appears.
2. Click View Search Result Attributes to display the View Search Result Attributes page.
3. Click Translate.

Note: The Translate button appears only if more than one language has been installed.

The Summary of Search Results Attribute Display Names page appears. Existing display names in all the locales are listed on this page. Display names that have not been configured for a language are marked as Not Configured.

4. Click Modify to configure a display name for a language.
The Configure Search Results Attribute Display Names page appears.
5. Click the language for which you want to configure a display name.

6. Enter the name in the Display Name field.
7. Click Save to save your changes (or Cancel to exit without saving your changes).

Adding Auxiliary and Template Object Classes to a Tab

You can use auxiliary object classes as mix-ins with structural object classes. For instance, if you have an auxiliary class for a person, and the auxiliary contains an attribute for the person's badge number, you might want to associate this auxiliary class with your structural object class. When you configure the User Manager, Group Manager, and Organization Manager applications, the more object classes you have at your disposal, the more information you can configure for users of those applications.

Note: You cannot remove an auxiliary object class you have added to a User Manager or Organization Manager tab. In Group Manager, under Configure Group Types, you can remove an auxiliary class.

If you have created workflows as described in “Chaining COREid Functions Into Workflows” on page 171, there are issues when associating an auxiliary object class with a tab:

- If the tab has associated workflows with pending requests, you cannot attach an auxiliary object class.
- If the auxiliary object class you are attempting to attach has any required attributes, you must edit all associated workflows to include those attributes.

You can also associate template objects with a tab. This is required if you plan to configure a workflow that makes use of the template object.

To add an auxiliary or template object class to a tab

1. From the COREid System Console, click User, Group, or Org. Manager Configuration > Configure Tab.

The Configure Tab page appears.

2. Click the link to the tab to display the View Tab page.
3. Click Modify to display the Modify Tab page.
4. In the Object Class(es) box, select an auxiliary or template object class or classes to associate with the tab.

These are added to the list on the left of the selection box when you save your changes.

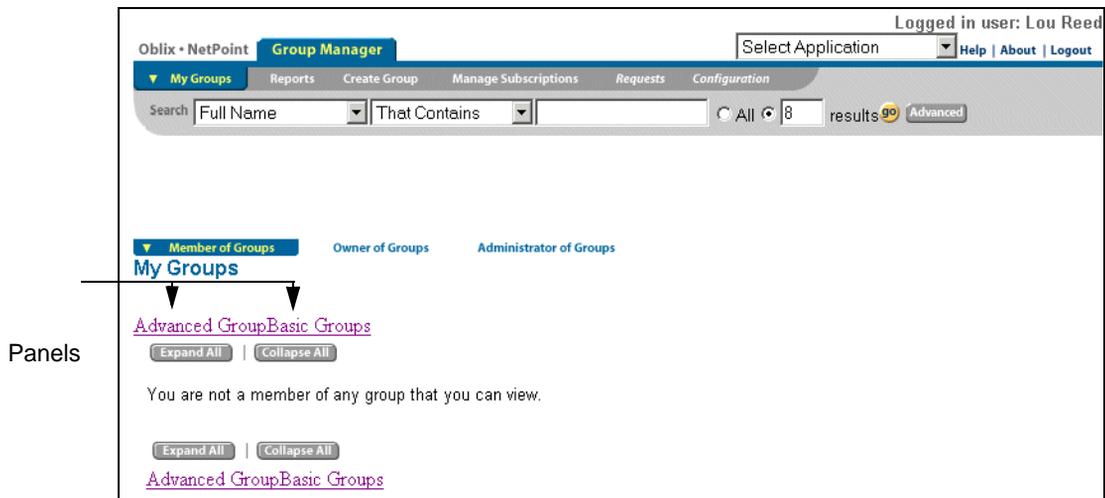
5. Click Save.

Adding Special-Purpose Object Classes to a Group Tab

Use Configure Group Types to associate auxiliary object classes with the Group Manager. The COREid System provides three LDAP group types that you can associate with the Group Manager tab:

- **groupOfUniqueNames**—Allows you to create named groups and define group owners.
- **oblixAuxLocation**—Allows to you configure locations.
- **oblixAdvancedGroup**—Allows you to configure attributes for subscribing members to groups.

The following shows the Group Manager application with the My Groups page selected. My Groups has been configured with multiple group type panels.



When you create a Group Type panel, the attributes from the associated object class are available in the Group Manager user application.

Configuring Group Manager Tab Options

Use the Group Manager Options feature to select what users see in the My Groups and View Members Profile pages of the Group Manager application. This feature allows you to turn off expensive operations. This can be useful if you need to enhance COREid performance.

To select what users see in My Groups and View Member Profiles

1. From COREid System Console, click Group Manager Configuration > Configure Group Manager Options.

The Configure Group Manager Options page appears.

2. Click Modify to display the Modify Group Manager Options page.

Modify Group Manager Options

This page shows the current settings of the various options for the Group Manager. You can change these settings and click Save. After that, your changes are reflected in the User application side of Group Manager.

Option	Value
Show static groups (in My Groups page)	<input type="checkbox"/>
Show nested groups (in My Groups page)	<input type="checkbox"/>
Show dynamic groups (in My Groups page)	<input type="checkbox"/>
Show groups you are a member of (in My Groups page)	<input type="checkbox"/>
Show groups you are an owner of (in My Groups page)	<input type="checkbox"/>
Show groups you are an administrator of (in My Groups page)	<input type="checkbox"/>
Show static user members of this group (in View Members page)	<input type="checkbox"/>
Show nested user members of this group (in View Members page)	<input type="checkbox"/>
Show dynamic user members of this group (in View Members page)	<input type="checkbox"/>
Allow users to override the defaults through URL parameters	<input type="checkbox"/>

This table below describes each option.

Option	Description
Show static group	Displays or hides groups consisting of individual members. Applies to the My Groups page.
Show nested groups	Displays or hides groups containing individual members and other groups. Applies to the My Groups page.
Show dynamic groups	Displays or hides groups with members that are determined by a filter. Applies to the My Groups page.
Show groups you are a member of	Displays the Member of Groups attribute on the My Groups page. You must also enable the Show static group, Show nested group, and Show dynamic group options to enable this function.
Show groups you are an owner of	Makes the Owner of Group attribute available on the My Groups page. You must also configure an attribute to be a Group Owner semantic type to use this feature.
Show groups you are an administrator of	Makes the Administrator of Group attribute available on the My Groups page. You must also configure an attribute to be a Group Administrator semantic type to use this feature.
Show static members of this group	Applies to the View Members page. You must configure an attribute to be a Group Static Member semantic type to use the static membership feature.
Show nested members of this group	Applies to the View Members page.
Show dynamic members of this group	Applies to the View Members page. You must configure an attribute to be a Group Dynamic Member semantic type to use the dynamic membership feature.
Allow users to override the defaults through URL parameters	Specifies whether or not the user can enter URL parameters to customize the Group Manager display options. Applies to the View Members page and My Groups page.

3. Select each option you want to apply to Group Manager.
4. Click Save.

Deleting a Tab in Organization Manager

If you have more than one tab in Organization Manager, you can delete a tab.

To delete a tab

1. From the COREid System Console, click Org. Manager Configuration > Configure Tabs and click the link to the tab.

The View Tab page appears. If you have more than one tab defined for Organization Manager, a Delete button appears on this page.

2. Click Delete.

You are prompted to confirm your decision.

3. Click OK to delete the tab and all associated information.

Ordering the Tabs in Organization Manager

You can change the order in which tabs appear in the Organization Manager when there is more than one tab listed.

To order the tabs in the Organization Manager

1. From the COREid System Console, click Org. Manager Configuration > Configure Tabs

2. Click the Order Tabs button below the list of tabs.

The Order Tabs page appears listing Tab 1, Tab2, and so on. Beside each tab number is a drop-down list that contains the names of existing tabs. .

3. Use the drop-down list beside each tab number to specify the order you would like.

For example:

Tab 1: Site

Tab 2: Location

4. Click Save.

Configuring Tab Profile Pages and Panels

A *profile page* is a Web page that shows information about an object in a COREid application. For example, when you search for information about a user in the User Manager, a profile page for that user is displayed. The profile page may contain data such as the user's

- Name
- Address
- Department
- Manager
- Phone number
- Email

The information on a profile page is based on objects and attributes in the LDAP directory that the COREid System communicates with, or it can be based on information in an object template file.

You can assemble profile pages from a collection of *panels*. For example, the profile page for a person may contain panels for personal, location, and project information. If you have configured an object template file for provisioning purposes, you may want to place the attributes from the template file on one particular panel.

Users can display profile pages in one of two ways:

- A panel view organizes the data on the profile page into panels.
- A page view organizes the data on the profile page into one long list.

Use of LDAP and Template Objects on a Panel

When you configure LDAP attributes on a panel, the attribute labels and values are shown on the profile pages that use the panel. In contrast, template attributes do not actually appear on the profile page. Template attributes only appear on Modify Profile pages, and then only if you have defined a workflow that uses the attributes.

See “Provisioning External Applications from COREid” on page 253 for details.

Configuring the Header Panel

The header panel appears at the top of a profile in the User Manager or Organization Manager. The header displays attributes with the semantic types of Full Name, Title, and Photo from the structural object class for the tab. You can turn the header off so that it is hidden from a user identity profile page.

Here is a sample header panel for a user:



Note: You can configure only LDAP attributes from the structural object class for a tab in header panels.

To configure the header panel

1. From the COREid System Console, click User or Org. Manager Configuration > Configure Tab.

The Configure Tab page appears. The Organization Manager may have multiple tabs.

2. Click a tab link, then click the View Object Profile button.
3. Click Configure Header, which is listed across the top of the page.

The Configure Header Panel page displays the attributes that appear in the Profile header. For example, Map Image, Location Name, Location Title.

4. Click the Modify button, then select each attribute to appear in the header panel.
5. If you want to display the Header Panel in user profiles, click Show Header Panel in User Manager.
6. Click Save.

Viewing Panels

You use panels to display a set of attributes on the User Manager, Group Manager, and Organization Manager pages.

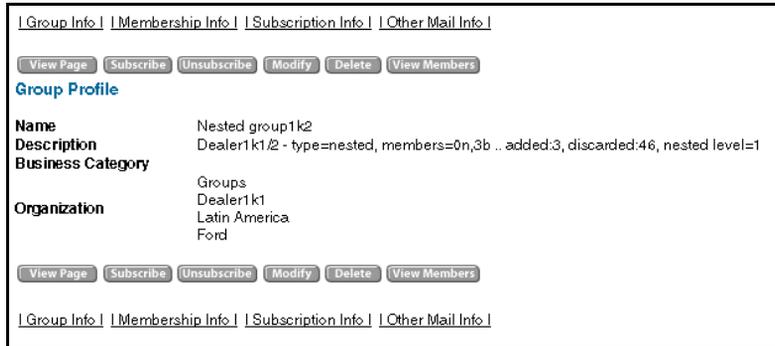
To view a panel

1. From the User, Group, or Organization Manager, conduct a search for a user, group, or organization object.
2. Click a link for a retrieved object.

The profile page for that object appears.

If the application displays the profile in a page view, click the View Panels button.

In the following example, the panels are Group Info, Membership Info, Subscription Info, and Other Mail Info. These panels are used in a group profile in the Group Manager application:



The following table shows some examples of panels for a user profile:

Table 15 User Profile Panel Attributes

Panel	Attributes
Telecommunications	Telephone number Fax number Cellular phone number
Location	Room Floor number Building number

Table 15 User Profile Panel Attributes

Panel	Attributes
Personal	Organization name Type Manager

Before configuring a panel, be sure the object class for the attribute that you want to place on the panel is configured with the appropriate object class type. See “Object Class Types” on page 64 for details.

Adding, Modifying, Localizing, and Deleting a Panel

You can create panels using the attributes configured during setup and when you performed the tasks described in “Making Schema Data Available to NetPoint” on page 59. You can use an attribute once per panel, and you can use the same attribute in more than one panel.

Note: You probably would want to configure one or more LDAP attributes or a combination of LDAP and template attributes on a panel. Since template attributes appear only in the context of workflow execution, a panel that consisted only of template attributes would appear to be empty.

If you have configured COREid for more than one language, you can view or modify the panel fields for each language.

You can localize display names for the following panel fields:

- Panel Label
- Description
- Attributes
- Mouse Over Message

To create or add a panel

1. From in the COREid System Console, click User, Group, or Org. Manager Configuration > Configure Tab.
2. Click the link to the tab.
The View Tab page appears.
3. Click View Object Profile.
4. Click the appropriate button at the top of the page:
 - For the User Manager and Organization Manager, click Configure Panels.

- For the Group Manager, click Configure Group Profile Panels.
The Configure Panels page appears. Currently defined panels are displayed.

5. Choose an operation:

- If you want to add a panel, click Create.
- If you want to modify a panel, click a panel link and then click Modify.
- If you want to delete a panel, click a panel link and then click Delete.

If you selected Create, the Create Panel page appears.

Create Panel

Panel Label

Description

Attributes

—	<input type="text"/>

Title Image

Tab Image

Depressed Tab Image

Tab Image (Bottom)

Depressed Tab Image (Bottom)

Mouse Over Message

Panel information is complete.

6. Edit the fields.

The Modify Panel page is similar to the Create Panel page. In both pages, the following fields are available:

Table 16 Panel Fields You Fill In

Label	Description
Panel Label	A name for this panel in the user application. This name can be localized.
Description	Text displayed in the View Panel page. This text can be localized.

Table 16 Panel Fields You Fill In

Label	Description
Attributes	<p>Attributes selected from the drop-down lists. If you need additional attribute fields, click Add at the right side of the page. Note that if you select template attributes, the attribute label will not appear on this panel. Template attributes are only displayed in the context of a workflow.</p> <p>These Attributes can be localized.</p>
Title Image	<p>You can view a user profile as a tab-separated page or as a single page. The Title Image is a GIF image that is used for the panel title when viewing a profile as a single page. The GIF must be stored in</p> <p><i>WebPass_install_dir/identity/oblix/lang/langTag/style0</i></p> <p>where <i>WebPass_install_dir</i> is the directory where you installed WebPass and <i>langTag</i> is the folder that contains the specific language that you are using.</p> <p>Enter the name of the GIF file, not the path. A Title Image can be modified as described in “Configuring Styles for COREid Applications” on page 266.</p>
Tab Image	<p>You can view a user profile as a tab-separated page or as a single page. The Tab Image is a GIF image that is used when viewing a profile as a tab-separated page. The Tab Image usually matches the Panel Label. Until you define a Tab Image, the Panel Label appears as a link on user profile pages. Clicking the link or the Tab Image opens a panel. The (Bottom) version is displayed at the bottom of user profile pages.</p>
Depressed Tab Image	<p>The image used when a user clicks a panel tab in a user profile.</p>
Mouse Over Message text	<p>Pop-up text that is displayed when the user positions the cursor over the Tab Image. This text can be localized.</p>

7. When this panel is ready for use, select **Panel information is complete** at the bottom of the page.
8. Click Save.

Note: Selecting Panel information is complete saves the panel definition, but a user’s ability to see the contents of a panel is governed by read permissions. The options are described in “Allowing Users to View and Change LDAP Data” on page 126.

To view or modify a panel

1. From the User, Group, or Organization Manager, click Configure Tab.
The existing tabs appear on the page.

2. Click a link to view its details.
The View Tab page appears.
3. Click the View Object Profile button.
The Configure Profile page appears.
4. Click Configure Profile Panels at the top of the page.
The appropriate Configure Panels page appears. Links for each of the configured panels are displayed on the page.
5. Click a panel link to view its details.
6. Click Modify to display the Modify Panel page.
7. Modify the information as needed.
8. Click Save to save your changes (or Cancel to exit the page without saving).

To localize a panel

1. From the User, Group, or Organization Manager, click Configure Tab.
The existing tabs appear on the page.
2. Click a tab to display the View Tab page.
3. Click the View Object Profile button to display the Configure Profile page.
4. Click Configure Panels to display links for each of the configured panels.
5. Click on a link to display the View Panel page.
6. Click Translate.

Note: The Translate button appears only if more than one language has been installed.

The Summary of Panel Display Names page appears. This page displays all configured language-specific display names for the following fields:

- Panel Label
- Description
- Attributes
- Mouse Over Message

Display names that have not been configured are marked Not Configured.

7. Click Modify to create or modify a display name.
The Configure Panel Display Names page appears. This page contains fields for the panel display names and links for all the installed languages.
8. Click the language of your choice.

9. Enter the display name in the appropriate field.
10. Click Save to save your changes (or Cancel to exit the page without saving).

Ordering the Panels

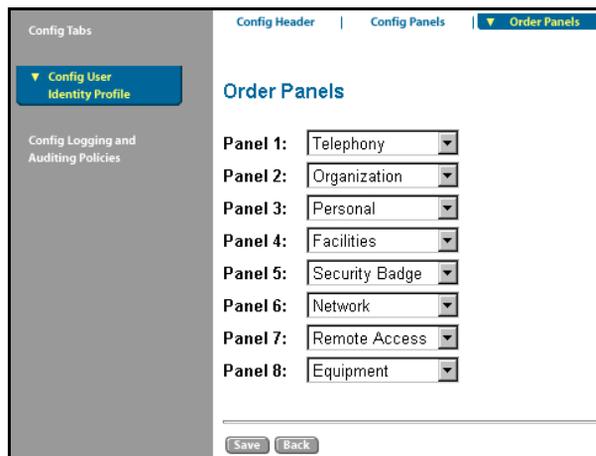
Panels appear in a particular order on a profile page. You can change the order in which they appear in the Group Manager.

To change the order in which panels are displayed

1. From the COREid System Console, click Group Manager Configuration.
2. Click Configure Group Types, then click the Order Group Type Panels at the top of the page.

Note: You can also select User Manager Configuration, Group Manager Configuration, or Organization Manager Configuration then select Configure Tab > *link* > View Object Profile > Order Panels. In the Group Manager Configuration, the option at the top of the page is Order Group Profile Panels.

The Order Panels page appears.



The screenshot shows the 'Order Panels' configuration page. The left sidebar contains 'Config Tabs' with 'Config User Identity Profile' selected. The main content area has a breadcrumb trail: 'Config Header | Config Panels | Order Panels'. Below the breadcrumb, the title 'Order Panels' is displayed. There are eight rows, each labeled 'Panel 1' through 'Panel 8', with a corresponding drop-down menu. The selected values are: Panel 1: Telephony, Panel 2: Organization, Panel 3: Personal, Panel 4: Facilities, Panel 5: Security Badge, Panel 6: Network, Panel 7: Remote Access, and Panel 8: Equipment. At the bottom of the form, there are 'Save' and 'Back' buttons.

3. Use the drop-down lists beside each panel number to identify the name of the panel to display.
4. Click Save.

Viewing Group Type Panels

Group Type panels allow you to organize attributes on the My Groups tab. For example, if you have configured `groupOfUniqueNames` as a structural object class and `oblixAdvancedGroup` as an auxiliary class, you can organize attributes from these classes on the My Groups tab by creating Group Type panels.

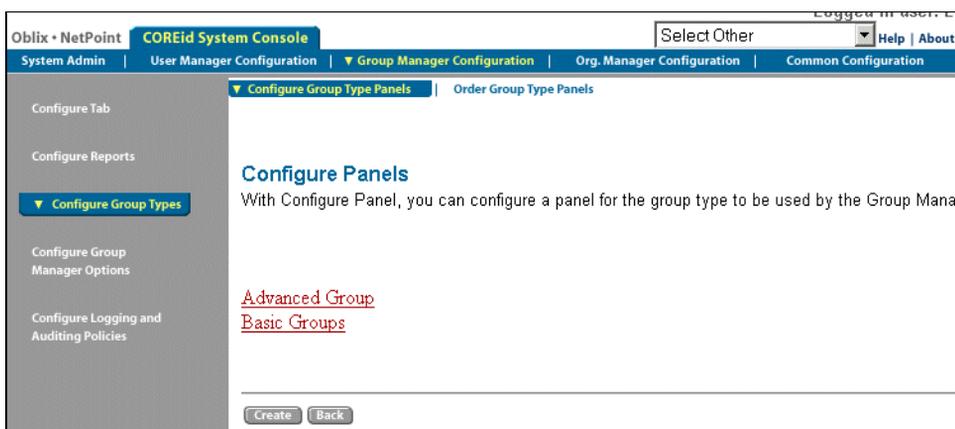
Note that Group Type panels are reserved for LDAP attributes. You should not configure *template* attributes on a Group Type panel.

Each object class identified as a Group Type (as described in “Object Class Types” on page 64) in NetPoint can be associated with a Group Type panel.

To view Group Type panels

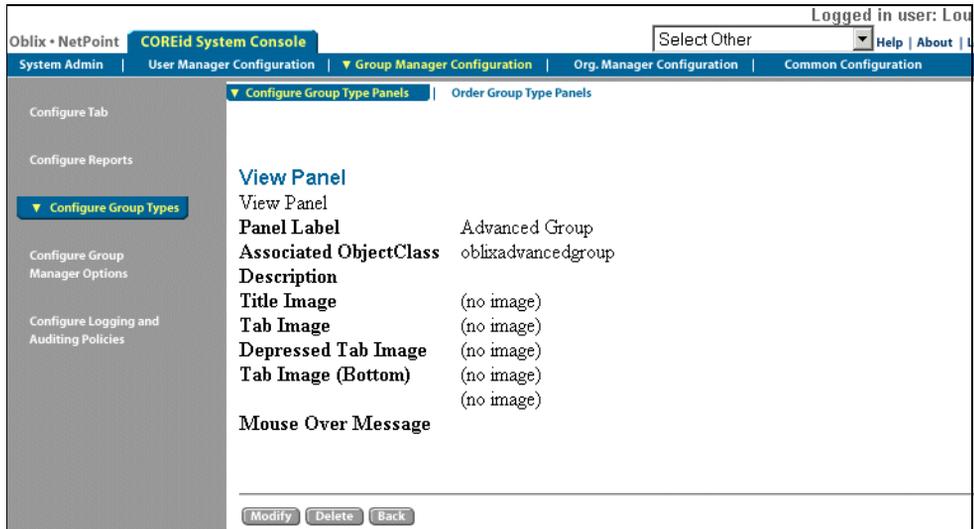
1. In the COREid System Console, click Group Manager Configuration > Configure Group Types > Configure Group Type Panels.

The Configure Panels page displays a list of configured Group Type panels.



2. Click its link to view a Group Type’s settings.

The View Panel page appears showing the settings for the selected panel.



Adding, Modifying, Localizing, and Deleting a Group Type Panel

You must configure a Group Type panel to organize the attributes for a group object class. At least one panel should be created for the group structural object class. This enables you to view groups that contain only the group structural object class attributes on the My Groups profile page.

If you have installed and configured multiple languages, you can localize display names for the following panel fields:

- Panel Label
- Description
- Mouse Over Message

To add, modify, or delete a Group Type panel

1. From the COREid System Console, click Group Manager Configuration > Configure Group Types.

The Configure Group Types page displays a list of Group Types.

2. Click Configure Group Type Panels to display the Configure Panels page .
3. Choose an operation:
 - To add a Group Type panel, click Create.
 - To modify an existing panel, click a panel link and from the View Panel page click Modify.

- To delete an existing panel, click a panel link and from the View Panel page click Delete.

If you clicked Create, the Create Panel page appears.

4. Select the object class to associate with the Group Type.

Note: Select only auxiliary object classes that extend the group structural object class or are attached to the group structural object class in the schema. Only configured auxiliary classes can be selected from this page. For more information, see “Adding Object Classes” on page 69.

5. In the remaining fields, enter values as described in “Panel Fields You Fill In” on page 117.
6. Select the box beside Panel information is complete.
7. Click Save.

Note: Selecting Tab information is complete saves the panel definition, but a user’s ability to see the contents of a panel is governed by read permissions, as described in “Allowing Users to View and Change LDAP Data” on page 126.

To localize panel display names

1. In the COREid System Console, click Group Manager Configuration > Configure Group Types > Configure Group Type Panels.
2. The Configure Panels page displays a list of configured Group Type panels.
3. Click the panel for which you want to configure display names.
The View Panel page appears.
4. Click Translate.

Note: The Translate button appears only if more than one language has been installed.

The Summary of Panel Display Names page appears. This page lists all the configured display names for the following fields:

- Panel Label
- Description
- Mouse Over Message

Display names that have not been configured for a particular language are marked as Not Configured.

5. Click Modify.
The Configure Panel Display Names page appears.
6. Click the language for which you want to configure display names.
7. Enter the display names for the panel fields.
8. Click Save to save your changes (or Cancel to exit the page without saving your changes).

Modifying and Localizing Attributes Displayed on a Panel

The attributes you configure through common configuration pages are used in each application using that object class. See “Making Schema Data Available to NetPoint” on page 59 for details. For instance, through common configuration you can set the display name for the cn attribute to be Full Name. This is what appears on a user Profile page. If you then configure the cn attribute to display as Legal Name from the User Manager configuration screen, it is displayed by default as Legal Name on the user Profile page.

You can also localize display names of attributes that are displayed on a panel. This allows you to present attributes in the user’s native language. See “Configuring

Multiple Languages for NetPoint” on page 272 for information on managing multiple languages.

Note: The only way to change the display type or semantic type of an attribute once it has been assigned to a panel is to delete and then re-create the panel.

To modify attributes specific to the User, Group, or Organization Manager

1. From the COREid System Console, click User, Group, or Organization Manager.

2. Click Configure Tab.

The Configure Tab page appears. There may be multiple tabs for the Organization Manager.

3. Click the link for the tab.

The View Tab page appears.

4. Click Modify Attributes.

The Modify Attributes page appears.

Details on modifying an attribute are provided in “Configuring Attributes” on page 81.

To localize attribute display names

1. From the COREid System Console, click User, Group, or Organization Manager.

2. Click Configure Tab.

The Configure Tab page appears. There may be multiple tabs for the Organization Manager.

3. Click the link for the tab.

The View Tab page appears.

4. Click Translate.

Note: The Translate button appears only if more than one language has been installed.

The Summary of Attribute Display Names page appears. This page lists all configured attribute display names for all languages. Display names that have not been configured are marked Not Configured.

5. Click Modify.

The Configure Attribute Display Names page appears. This page lists display name fields for attributes and links for the installed languages.

6. Click the language for which you want to configure display names.
7. Enter the name in the Display Name field.
8. Click Save to save your changes (or Cancel to exit the page without saving your changes).

Allowing Users to View and Change LDAP Data

You can think of configuring objects and attributes and assembling attributes into panels on application tabs as being like playing with building blocks. Once you have arranged your building blocks, you can determine who is allowed to play with them.

You must configure the COREid System to allow people to search for and view the LDAP attributes you have configured on the application panels. . To do this, you:

- Determine the level of the directory tree that users are permitted to search.
- Set View and Modify permissions for specific attributes in the directory tree.

Note: The following section discusses setting the searchbase as a method of configuring view and modify permissions. The searchbase refers to searching the LDAP directory tree. Template attributes are not relevant to setting a searchbase. To give users the ability to enter values for template attributes, the users must be participants in a workflow where these attributes are used. See “Chaining COREid Functions Into Workflows” on page 171 for details.

About the Searchbase

A searchbase is a branch in the directory tree, or it can be the top node of the tree. At installation time, you select the default searchbase. The default searchbase is the node in the directory tree under which all user data is stored and the highest possible base for all user data searches. The searchbase determines the part of the directory tree that is available to a user during a search. You must set a searchbase for each structural object class configured for the COREid System before a user can view its entries. You can set multiple searchbases per structural object class.

When you set a searchbase, you determine who can search what (an object class, at a particular level of the directory tree), optionally using a search filter.

Before setting a searchbase you need to determine the following:

- What object class (users or groups) do I want to set the searchbase for?
- Where will the search begin?
- Who can search there?

For example, you can configure one searchbase for employees and another for customers to ensure that customers cannot see employee information.

As another example, if two competing suppliers provide you with parts, you can set the searchbase so that users from each supplier can view only their own portion of the DIT.

Note: You set the searchbase from the User Manager application. This is the end user application rather than the User Manager Configuration function. You also need to configure read permissions for your group profile pages for the group class.

Guidelines for Setting the Searchbase

When you set a searchbase, you have the option to define a filter to identify what branch of a searchbase a logged-in user can view. If your directory tree is particularly flat, so that selecting a node does little to filter the searchbase, the filter feature helps narrow searches. The filter is also useful if your directory tree has a large number of branches, for instance, if you have 10,000 dealerships, you probably want to narrow down searches within the dealerships.

However, a filter can affect performance if it yields a large number of entries. Instead of using a searchbase filter, you can set read permissions for the class attribute, as described in “Selecting a Class Attribute” on page 67. The class attribute is used for attribute access and to link search results to a Profile page.

For example, suppose you remove the resource filter from the searchbase, allowing the role of Anyone to access the person object class. Instead of setting the searchbase, you define read permissions for the class attribute, using a rule to specify who can access this attribute. This can reduce the number of directory searches that NetPoint conducts. See “Setting and Modifying LDAP Attribute Permissions” on page 138 for details.

Note: You can set several searchbases for the same user or group if specific users need to access different parts of the directory tree. For example, if employees need to search both the employee and the customer branches of the tree, you can define searchbases for employees and for customers, and give employees permission to view both. However, be sparing when configuring multiple searchbases for a particular object class. Where possible, define read and write permissions for attributes instead. Multiple searchbases for the same object class can degrade performance.

If You Need to Modify a Searchbase

If you change the levels being searched in the directory tree or if you change the search attribute, you cannot directly modify a searchbase. If you attempt to do so, NetPoint treats the modified searchbase as a newly defined searchbase. The only way to modify a searchbase is to delete it and create a new one.

Note: You can modify a searchbase if the changes are other than those described above.

Setting the Searchbase

The following procedure describes the steps for setting the searchbase.

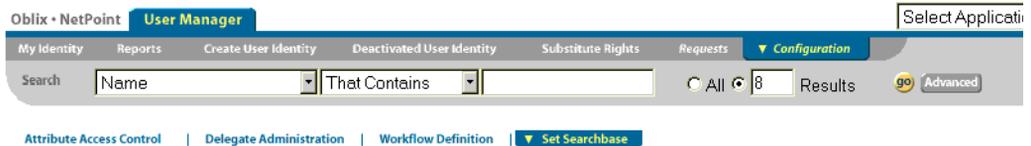
To set the searchbase

1. In the User Manager application, click the Configuration command on the top menu bar.

The Configuration page appears.

2. Click Set Searchbase.

The Set Searchbase page appears, as shown in part below.



Set Searchbase

Set the searchbase for a particular organization or person. This will localize access to ensure security

1) Objectclass	genSiteOrgPerson
2) Searchbase Domain	<input type="text" value="o=company,c=us"/>
Filters	(objectClass=*)
Add Filter	(objectClass=*)
3) Target Domain	<input type="text" value="o=company,c=us"/>

3. In the Object Class drop-down list, select an object class.

The object class you select defines what is being searched. For instance, to set a searchbase for widgets, you would select the widget object class.

The Searchbase Domain box indicates the top node for the search. The field beneath the Searchbase Domain box is where you enter or edit information.

4. In the field under the Searchbase Domain box, specify the part of the directory tree where the search for the object may be conducted.

For instance, if you want to define a searchbase for widgets in the Manufacturing Department, you might select the Manufacturing branch of the searchbase.

Selecting the top level of the directory tree indicates that the entire domain is available for searches. You can refine the searchbase by selecting a node further down the tree or by entering a filter. For example, to restrict searches to North America, you could select the top node and enter `region=North America` as the filter. This example assumes there is a branch called North America in your directory tree. See “Usage of Rules and Filters” on page 89 for details on writing a filter.

The Filters box indicates the current filters for the search. You use the Add Filter field, beneath the Filters box, to enter another filter.

5. **Optional**—In the Add Filters field, enter another filter.

6. Click Save.

The new filter displays in a field below the previous filter.

Users and groups permitted to search this portion of the directory tree are defined in the next panel.

7. Specify the user or group that is permitted to search this portion of the directory tree.

For example:

- **Target Domain**—Any user object in the tree under the node you select.

Do *not* use full LDAP URL while specifying the filter for target domain (or workflow domain) while creating the workflow. Only the LDAP filter is expected. For example, `cn=Shutterbug Canavan` is expected rather than `ldap:///ou=Partners,o=Company,c=US??sub?(cn=Shutterbug Canavan)`.

- **Role**—The role of the users.

If you want to give this right to everyone whether they have logged in or not, select **Anonymous**.

If you want to give this right to anyone who has logged in to the User Manager, Group Manager, or Organization Manager, select **Anyone**.

Note: Anonymous access is used only in the Self Registration function in User Manager and Organization Manager. Also, anonymous access applies only to display type attributes (a check box, radio button, or drop-down list) that are configured as a Rule. For example, suppose you configure the ou attribute as a drop-down list display type with a rule that uses the LDAP filter (objectclass=organizationalunit). To configure this attribute for self registration, you would access the Organization Manager tab for organizationalUnit, configure attribute access for the class attribute (as described in “Setting and Modifying LDAP Attribute Permissions” on page 138), and grant Anonymous access.

- **Rule**—Any person you specify with an LDAP filter. Click **Build Filter** and use the **Query Builder** to create a rule. See “Writing LDAP Filters Using Query Builder” on page 132 for details.
- **Person(s)**—Any person you select. Click **Select User** and use the **Selector** to choose individuals.
- **Group()**—Any group you select. Click **Select Group** and use the **Selector** to choose one or more groups.

To copy users and groups from one searchbase to another, click **Copy**, click **Reset**, select another **Searchbase Domain** and **Target Domain**, and click **Paste**. The users and groups appear in their respective boxes.

Note: If you specify users by more than one means (for instance, by a rule and by selecting individual users), both methods apply. The only exception is when **Anyone** is selected. **Anyone** supersedes all other methods.

8. Click one of these buttons to take the appropriate action:

- **Save**—Save and implement changes.
- **Reset**—Clear all selections.
- **Delete**—Clear all rule, group, and user specifications.
- **Report**—Generate a report summarizing the configured searchbases.

If You Set a Searchbase for a Group

You can set the searchbase for the groupOfUniqueNames object class and select the groups for which you are defining the searchbase. Before people in the group can view entries in a searchbase for a group, you need to configure read

permissions for your group profile pages for the group class, as described in “Setting and Modifying LDAP Attribute Permissions” on page 138.

Setting Up Disjoint Searchbases

A disjoint searchbase is a searchbase that supplements the one you selected when you set up the COREid System. You create a disjoint searchbase to identify an additional LDAP directory tree under which user data can exist.

You can add multiple disjoint searchbases to a domain.

To add a disjoint searchbase for a disjoint domain

1. From the COREid System Console, click System Admin > System Configuration > Configure Directory Options.
2. Click the Directory Server link.
3. Add a disjoint searchbase in the Disjoint_domain field and click Save.
4. From the COREid System Console, click User Manager Configuration.
5. From the side navigation bar, select Configure Tab.

The Configure Tab screen appears.

6. Select the tab link.
7. Click Modify.
8. Make sure there is no value in the Tab Searchbase field.
9. Save your changes, if necessary.

To delete a disjoint searchbase

1. Disable all database agents that use this searchbase.

Note: You should also remove all access control policies for a disjoint searchbase before deleting it.

If there are policies defined for the deleted searchbase, a user who has this searchbase on this node will be able to create a filter using Query Builder whose base is this searchbase.

2. From the COREid System Console, click System Admin > System Configuration > Configure Directory Options.
3. Click the Directory Server link.
4. Remove the information in the Disjoint_domain field, then click Save.

Writing LDAP Filters Using Query Builder

The Query Builder enables you to write LDAP filters when you perform activities such as setting the searchbase.

NetPoint enforces a limit of 20 hits per query. This applies to both Selector and Query Builder. If you perform a search or query that results in more than 20 hits, you receive truncated results. For instructions on changing the search limit, refer to the `cookieBustLimit` parameter in the *NetPoint 7.0 Customization Guide*.

You access the Query Builder function from the Build Filter button. For example, this function is available when setting a searchbase. See “Setting the Searchbase” on page 128 for details.

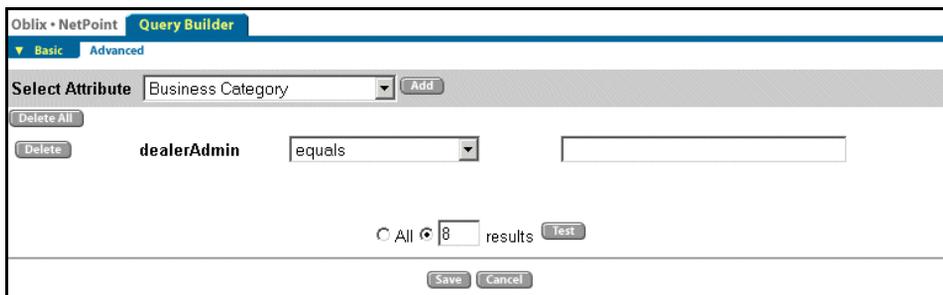


Note: If you choose the Is Present or Is Not Present operator when building a query, the value specified for the display type is not taken into consideration, since the filter that is used is a presence filter.

To use the Query Builder

1. From the Set Searchbase page, locate and click the Build Filter button.

The Query Builder page appears. By default, the Basic query page is displayed.



2. In the Attribute drop-down list, select an attribute you want to use as search criteria.

For example:

Admin

3. Click Add.

The attribute is added to the filter.

4. From the drop-down list beside the new attribute, select a matching method.

For example:

greater than or equals

The available methods depend on the attribute. See “Methods for Retrieving Matches” on page 133 for details.

5. In the field beside the method, select or type the query string.

For example:

January 22 2003

6. **Optional**—Click Add to add other attributes.

7. From the drop-down list to the left of the attribute, select the relationship between attributes:

- **And**—Results must match criteria in all rows.
- **Or**—Results can match criteria in one row.

For example, you can search for everyone with the Administrator attribute and a start date after (greater than) January 22, 2003.

8. Click Test to test your filter.

If too many or too few results are received, make your criteria more or less restrictive.

9. Click Delete next to an attribute to remove it from the filter (or click Delete All to delete all attributes).

10. Click Save.

When you Save, the filter appears in the previous page below the Build Filter button.

Note: If you receive a Bad Request message when you save, your query string is too long for your browser. Browsers handle the filters as URLs, and they generate an error if the query string exceeds their maximum URL length.

Methods for Retrieving Matches

The matching methods that you can select in the Query Builder depend on the display type of the attribute. For instance, the display type of an attribute may be a list or a set of radio buttons. See “Attribute Display Types” on page 78 for details. When you use the Query Builder to create a filter for an attribute with a display type that contains multiple values, for example, a list, the query returns a match even if only one value satisfies the filter.

When building a filter, you can select multiple values for an attribute in one row only if the attribute display type is a check box or a radio button.

The Query Builder uses the following matching methods:

Method	Description
equals	Results are an exact match of the value.
does not equal	Results do not include the specified value.
less than or equals	Results are less than or equal to the specified value. For example, specifying <code>k</code> for a full name query returns people whose name begins with a letter from A to K.
greater than or equals	Results are greater than or equal to the specified value. For example, specifying <code>k</code> for a full name query returns people whose name begins with a letter from K to Z.
less than	Returns any directory entry with a value that is less than the specified value. When filtering a text string, a value of less than returns entries that precede the specified value alphabetically. For example, specifying <code>k</code> for a full name query returns people whose name begins with the letters from A to J.
greater than	Returns any directory entry with a value that is greater than the specified value. When filtering a text string, a value of less than returns entries that follow the specified value alphabetically. For example, specifying <code>k</code> for a full name query returns people whose name begins with the letters from L to Z.
contains	Returns any directory entry that contains the specified string anywhere in the value of the entry. For example, an entry of <code>st</code> might return values of <code>street</code> or <code>best</code> .
does not contain	Returns any directory entry that does not contain the specified string anywhere in the value of the entry.
is present	Returns any directory entry that contains this attribute. For instance, if you select the <code>Administrator</code> attribute and the <code>is present</code> method, all administrators are returned.
is not present	Returns any directory entry that does not contain this attribute.
begins with	Returns any directory entry that begins with the specified value.

Method	Description
ends with	Returns any directory entry that ends with the specified value.
does not begin with	Returns any directory entry that does not begin with the specified value.
does not end with	Returns any directory entry that does not end with the specified value.
sounds like	Results approximate the sound of the specified value. Use this option if you are unsure of the spelling of your desired search object. Use phonetic spelling. For example, specifying kiero might return values for cairo. This option is not supported by Novell Directory Services.
does not sound like	Results display entries that do not approximate the sound of the specified value. Use your best phonetic spelling. This option is not supported by Novell Directory Services.

Building Advanced LDAP Filters Using QueryBuilder

Filters can work on multiple attributes and use logical operators such as And, Or, and Not.

To build a complex filter

1. In the Query Builder page, click the Advanced tab.
2. If you switch from Basic to Advanced, and you choose OK, you lose the current filter (click Cancel to keep the displayed filter).

The Advanced page appears.

Select Attribute equals

Select Separator

Constructed Visual Filter

LDAP Filter

All 8 results

If the Advanced page does not appear after you click the Advanced tab, the URL could be too long. The length of the URL is determined by the browser.

3. In the Select Attribute drop-down list, select the attribute you want to use as the search criteria.
4. In the associated drop-down list select a matching method, and in the associated text entry field add a query string.

See “Writing LDAP Filters Using Query Builder” on page 132 for details.

5. Click Add.

The attribute is added to the Constructed Visual Filter box.

6. You can perform the following optional steps:
 - To add to your LDAP commands, use the And, Or, or () buttons.
 - To remove an attribute from the Constructed Visual Filter box, select it, and click Delete (or Delete All to remove all attributes).
 - To modify an entry in the Constructed Visual Filter box:
 - Select the entry.
 - Make your changes to the query characteristics at the top of the page.
 - Click Modify.
7. Click Show LDAP Filter to view the filter you are building.

The LDAP string displays in the LDAP Filter box. You can edit the text in this box and click Update Visual Filter. For examples of LDAP filters, see “Static LDAP Search Filters” on page 90 and “Examples of Dynamic LDAP Search Filters” on page 92.

If you manually enter a very complex filter, the Constructed Visual Filter box may not be able to interpret it correctly. However, the filter will work correctly.

8. Click Test to view the results of your query.

NetPoint displays output that conforms to your filter.

9. Click Save to save and apply your filter.

If you receive a “Bad request” message when you click Save, your query string is too long for your browser. Browsers handle the filters as URLs, and they generate an error if the query string exceeds their maximum URL length.

About View and Modify Permissions

Until you configure permissions for an attribute, no users can see attributes displayed in the User Manager, Group Manager, and Organization Manager. For example you can allow all users to view employee work phone numbers in the User Manager, but allow only managers to view home phone numbers.

If you are a Master Identity Administrator or a delegated administrator with appropriate permissions, you can configure user permissions. By default, NetPoint Administrators specified during COREid Server installation have full access to all attributes. You can change the default by setting the `BypassAccessControlForDirAdmin` parameter to false in:

```
COREid_Install_Dir/identity/oblix/apps/common/bin
```

Setting and Modifying LDAP Attribute Permissions

The Attribute Access function lets you specify permissions that determine who can read and modify the values for each LDAP attribute. It also lets you create a list of users or groups to be notified when an attribute is changed. As with setting the searchbase, this functionality only applies to LDAP attributes. You configure permissions for template objects when you add participants to workflow steps. See “Chaining COREid Functions Into Workflows” on page 171 for details.

Users must have a searchbase defined as well as read permissions to be able to view an attribute. For instance, to be able to view the class attribute on the User Manger, Group Manager, or Organization Manager tab, a user must be a trustee of the appropriate searchbase domain for the class attribute, and they must have read permissions for this attribute.

To set or modify attribute permissions

1. In User, Group, or Organization Manager, click Configuration at the top of the page.

The Configuration page appears.

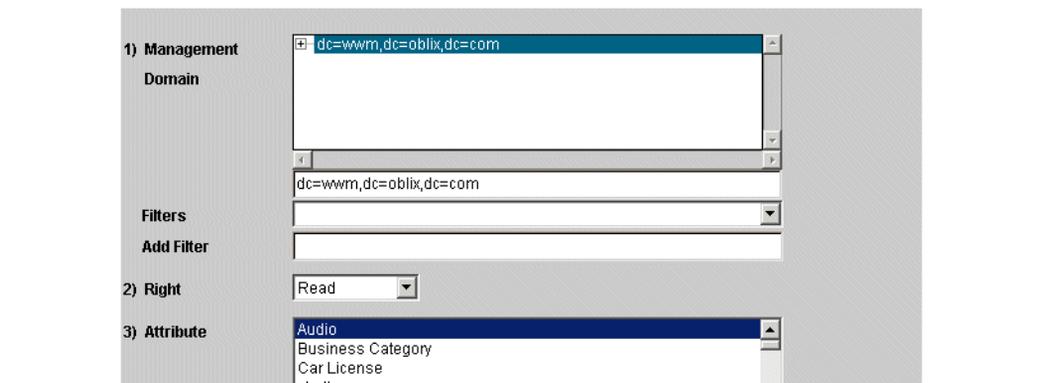
2. Click Attribute Access Control.

The Attribute Access Control page appears.



Attribute Access Control

Attribute Access Control allows an administrator to control who has read and write privileges on each attribute. It also allows a notification list to be specified when a change to an attribute is requested. The access control and notification can be set at any and all levels of an organization.



3. In the Management Domain box, specify the scope of the Directory Information Tree (DIT) that this permission applies to.

Initially, this field displays the searchbase set during product setup. This searchbase can only be changed by performing setup again. Selecting a lower level in the tree applies access control for that branch. For example, if you select the Full Name attribute, and then select a lower level department such as Sales, you are applying access control to all of the people in Sales with Full Name in their profile.

4. **Optional**—Use the Filters field to enter an LDAP rule to specify the objects and attributes more precisely.

A filter refines the attributes you are allowed to read or modify. If you do not use a filter, NetPoint uses `objectclass=*`.

Note: A filter is useful if your database design is particularly flat or has a particularly large number of branches.

Add the Filter in the Add Filter field. Once the configuration is saved, the filter is added to the Filters list. If you later want a different filter, you must delete the original searchbase and create a new configuration.

For more information on filters, see “Usage of Rules and Filters” on page 89.

5. Specify the Right:

- **Read**—Selected users can view the attribute and its value on a profile page.
- **Modify**—Selected users can change the attribute value. Note that you must confer read permissions for these users to be able to see the attribute value.
- **Notify**—Sends an email to the specified users when an attribute value is changed.

For example, you can give read and modify permissions to the Title attribute for a manager. Then you can set notification to be sent to the HR department when the value for this attribute is modified in a user profile. For details about email post-notification for a self registration step, see “Descriptions of Step Actions” on page 186.

6. In the Attribute box, select the attribute to associate with this right.

If you want to make multiple selections, see “Keys for Selecting Multiple Attributes” on page 141.

Note: If an attribute in your multi-select range has a different set of trustees, a pop-up error appears. This prevents you from inadvertently allowing access to incorrect trustees (participants).

7. Confer this right to one or more of the following:

Role—Assigns the right based on the user’s role. Any attribute with a data type of DN and a display type of Object Selector appears in the Role area. Self and Anonymous are shipped with the COREid System. Each application contains different roles, largely dependent on your configured attributes. For example, the User Manager may have the Manager role, but not any role based on the secretary attribute, depending on your configuration. Common roles include the following:

Anyone	All users who log into the User, Group, or Organization Manager can either view or modify the attribute at the selected level. For example, all logged-in users can view the phone number attribute at the specified level in the directory.
Anonymous	All users can view entries, whether they are logged in or not. Anonymous access is only used for self-registration.
Self	The user logged into the User Manager application can view or modify the attribute for his or her own identity, assuming the read and write permissions for attributes is high enough on the directory tree to include the user’s profile. For example, if you select Self to be able to view the Name attribute at the top level, then you, as a person logged into the User, Group, or Organization Manager, are able to view your name. But if you specify ou=Marketing as the level on the directory tree, and the user is not in Marketing, then you cannot view your name.
Manager	The user logged in to the User Manager application can either view or modify the attribute for their direct reports.
Secretary	If the user logged in to the User Manager is an administrative assistant, he or she can view or modify the attribute for the people they support.
Group Owner	The user logged in to the Group Manager can view or modify the attribute for the group that he or she is an owner of.
Group Administrator	The user logged in to the Group Manager can view or modify the attribute for the group that he or she administers.
Group Member	The user logged in to the Group Manager can view or modify the attribute for the group that he or she is a member of.

Rule—Click Build Filter and use the Query Builder to create a rule. See “Writing LDAP Filters Using Query Builder” on page 132 for details.

Person(s)—Click Select User and use the Selector to specify one or more users.

Group(s)—Click Select Group and use the Selector to specify one or more groups.

See “Evaluation of LDAP Attribute Permissions” on page 142 for information on the order for evaluating permissions.

8. Click Copy, click Reset, select a new attribute, and click Paste to copy users and groups from one attribute to another.
9. Click one of these buttons:
 - **Save**—Save and implement your changes.
 - **Reset**—Clear all selections.
 - **Delete**—Clear all rule, role, group, and user specifications.
 - **Report**—Generate a report of attributes and their access permissions in the domain.

Keys for Selecting Multiple Attributes

You can configure access control for multiple attributes at one time using the following keyboard combinations:

- **Ctrl + Home**—Selects all attributes above and including the highlighted attribute.
- **Ctrl + End**—Selects all attributes below and including the highlighted attribute.
- **Ctrl + Page Up**—Selects only attributes above the highlighted attribute.
- **Ctrl + Page Down**—Selects only attributes below the highlighted attribute.

Note: If an attribute in your multi-select range has a different set of trustees (participants), you receive an error. This prevents you from granting access to incorrect trustees.

Platform-specific key combinations are as follows:

Windows Browsers	<ul style="list-style-type: none">• To select multiple attributes, hold down the Ctrl key and select the attributes.• To select an attribute and all attributes preceding it, hold down the Ctrl+Shift+Home keys and select the attribute.• To select an attribute and all attributes following it, hold down the Ctrl+Shift+End keys and select the attribute.• To select an attribute and an arbitrary number of attributes following it, select it and press Shift+Down Arrow.• To select an attribute and an arbitrary number of attributes preceding it, select it and press Shift+Up Arrow.
Unix Browsers	<ul style="list-style-type: none">• To select multiple attributes, hold down the ESC key and select the attributes.• To select an attribute and all attributes preceding it, hold down the ESC+Shift+Home keys and select the attribute.• To select an attribute and all attributes following it, hold down the ESC+Shift+End keys and select the attribute.

Evaluation of LDAP Attribute Permissions

When you assign multiple methods for view and modify permissions, NetPoint evaluates the methods in this order:

1. Users
2. Roles
3. Groups
4. Rules (LDAP filters)

When NetPoint finds a match, it stops checking. For example, suppose you grant read permission for the Name attribute for User=Lou Reed, but you also have a rule that says (&!(cn=Lou Reed)) objectclass=*person object class*, which allows everyone except Lou Reed. Lou Reed has access because he is a User, which precedes Rule in the evaluation order. As another example, if you specified a rule denying access to the Human Resources department but used the people selector to specify an individual employee in Human Resources, the union of the rule and people categories would allow access to the specified employee.

Note: If you select the *Anyone* role, all users, roles, groups, and filters are superseded.

Examples of Configuring an Application

The following sections describe different scenarios for configuring an application. Separate examples are provided for the User Manager, Group Manager, and Organization Manger.

Displaying Photos in User Profiles

Photos appear in the header panel of a user profile. Users with self-service permissions on relevant attributes can manage their own photos.

There are two ways you can store photos in NetPoint:

- In an LDAP directory
- Referencing photos in a file system.

You cannot use both methods. All of your photos must be stored in either a directory or a file system.

Importing and Storing Photos in a Directory

When you want to store your photos or other images in a directory, place the photos on the COREid Server and use NetPoint to import the photos into the directory and to configure an attribute to be the photo attribute. You can create your own attribute, or you can use an attribute that already exists. This attribute must be defined as a binary type in your directory, and in NetPoint the attribute must be defined as a Photo semantic type with a GIF display type. The GIF display type supports GIF and JPEG formats, and other image file formats that are supported by your Web server.

Before associating a photo with a user's identity, be sure the photo's file name is based on the value of the attribute with the Login semantic type. For example, if your Login semantic type is assigned to the uid attribute, you would use the following file name conventions:

```
attribute_value_of_uid.gif  
or  
attribute_value_of_uid.jpg  
or  
attribute_value_of_uid.jpeg
```

If your login semantic type is something other than uid, use that instead for your file name. For example, if the Login semantic type is assigned to the email attribute, your photo file names must be the following:

attribute_value_of_mail.gif
or
attribute_value_of_mail.jpg
or
attribute_value_of_mail.jpeg

The file extension must be compatible with a graphic file format that your Web servers can support.

When NetPoint imports photos and images, it converts the files into Base64 format. This data becomes the value of the Photo attribute. NetPoint uses the Login attribute and the photo or image file name to determine which photo belongs to which user entry.

Steps for configuring NetPoint to use the photos are described in the following discussions.

To configure photos for importing to a directory

1. From the COREid System Console, click Common Configuration > Configure Object Class.
2. Select your person object class from the list.
3. Click Modify Attributes.
4. Configure the Photo attribute as follows:

Attribute—Photo

DisplayName—Photo

Semantic Type—Photo

Data Type—Binary

Attribute Value—This is always a single value attribute

Display Type—GIF Image

5. Save your changes.
6. In the User Manager, under Attribute Access Controls, assign Read and Write permissions to this attribute.

To import photos to the directory

1. From the COREid System Console, click System Admin > System Configuration > Import Photos.
2. Specify the path to the photos stored on the COREid Server.
3. Click Save.

This imports all of the GIF and JPEG images into your directory.

Referencing Photos in a File System

Another method for storing images and photos for user identities is to store the photos in a location other than the directory. This method is appropriate for GIF and JPEG images, and other image file formats that are supported by your Web server.

The COREid Server's WebPass must be able to access this location. You can name the photo or image file using any valid file name that the Web server recognizes and supports. Avoid using special characters such as spaces in the file name. The Web server may not recognize file names that use special characters.

To reference photos that reside in a file system

1. Click User Manager > Common Configuration > Configure Object Class.
2. Click the person object class in the Object Class list.
3. Click Modify Attributes.
4. Configure the Photo Path attribute as follows:

Attribute—Photo Path

DisplayName—Photo

Semantic Type—Photo

Data Type—String (case-sensitive)

Attribute Value(s)—Single or multi-valued

Display Type—GIF image URL

5. Assign read and write permissions for this attribute.
6. Store the images in GIF or JPEG format in the following directory:

WebPass_install_dir/identity/oblix/lang/*langTag*/style0

where *WebPass_install_dir* is the directory where WebPass is installed and *langTag* is the folder that contains the specific language you are using.

7. Enter the photo location URL in the User Profile Page for each user.

For example, if the image location is:

c:\NetPoint\WebComponent\identity\oblix\apps\lang\en-us\style0\user1.gif

you set the photo location to:

user1.gif

More than one GIF image can be displayed by setting the photo URL attribute to be multi-valued.

The Default Photo Image

NetPoint supplies a default photo image. This image is presented in case there is no photo image supplied for a user. The image is stored in CIMAGEdefaultphoto.gif in style0 on the COREid Server.

Enabling the Location Tab in Organization Manager

NetPoint provides a Location tab by default in the Organization Manager. This tab enables you to create maps and associate users or objects with locations on those maps.

Task overview: Enabling Location functionality

1. The Master Identity Administrator modifies the Location tab and adds location attributes to Profile pages for the User and Organization Manager applications.
2. The Master Identity Administrator configures access controls for location attributes.
3. The Master Identity Administrator or Delegated Identity Administrator configures workflows for creating a location. See “Chaining COREid Functions Into Workflows” on page 171 for details.
4. The Delegated Identity Administrator creates a new location and establishes the location’s hierarchy in relation to other locations, if applicable.
5. The Delegated Identity Administrator or user assigns a value for the location attribute for a user or object profile.

Any user with appropriate permissions can now view the user or object location.

The Right to Create Groups in Group Manager

You assign users the right to create a group when you define a Create Group workflow. Only users designated as participants in the workflow can create the group. See “Chaining COREid Functions Into Workflows” on page 171 for information about creating workflows.

A user can be assigned the right to modify a group type if the user is a participant in a Create Group workflow for that group type. The user must also have write access for the group type attribute. See “Adding Special-Purpose Object Classes to a Group Tab” on page 109 for information about group types. Also see “Setting and Modifying LDAP Attribute Permissions” on page 138 for information about assigning the modify right to the Group Type attribute.

If you run NetPoint with multiple Active Directory instances and use a dynamic filter to create a group, the filter attribute must be a multi-value attribute.

If you run NetPoint with the NDS directory, the users you select as members of the group are cleared from the page when you click Save. To prevent this from happening, go into the NDS directory and switch the order of the attributes so `uniquemember` is read first. Also make sure the `userCertificate` attribute comes before the NDS `userCertificate;binary` attribute.

End-User Scenarios

The following sections describe how an end user interacts with the Group Manager application once it has been configured:

- “Managing Group Members in Group Manager” on page 147
- “Searching for Group Members” on page 147
- “Customizing Search Results for Group Members” on page 149
- “Deleting Group Members” on page 150
- “Adding Group Members” on page 150
- “Managing Group Subscriptions” on page 151
- “Subscribing to Groups” on page 152

Managing Group Members in Group Manager

You can view and manage group members from the Group Profile page if the Master Identity Administrator selected a group-member attribute to display on the group profile page. See “Configuring Tab Profile Pages and Panels” on page 113 for more information.

If your group contains a large list of members, this can negatively impact system performance. The Master Identity Administrator can choose not to display group members on the Group Profile page. See “Configuring Group Manager Tab Options” on page 110.

You can also view and manage group members from the Manage Group Members page.

Searching for Group Members

The Manage Group Members page enables you to view the members of a group based on criteria that you provide. This page shows tables for:

- Static members
- Dynamic members
- Nested members

Search results are subject to searchbase and attribute access controls configured for the Group and User Manager applications. See “Setting and Modifying LDAP Attribute Permissions” on page 138 for details.

If a user does not have read access to the dynamic member attribute for a group, nothing appears in the dynamic member table and the following error message is shown, “You don't have read access for a dynamic member.”

In the nested members table, if the group contains dynamic nested groups and the user does not have read access to the dynamic member attribute for some of the nested groups, the dynamic members are not shown. In this case, no error message is displayed.

To view group members

1. In the Group Manager, click My Groups.
2. Conduct a search on groups and click the desired link.
The group profile appears.
3. Click Manage Group Members.

The screenshot displays the NetPoint Group Manager interface. At the top, there is a navigation bar with 'Obliv • NetPoint' and 'Group Manager' tabs. Below this is a search bar with 'Full Name' selected, 'That Contains' as the operator, and '8 results' shown. The main content area is titled 'Manage Group Members' and includes a 'Group Information' section with the following details: Full Name (IT Operations), Description (WWM Corporate), Organization, Organizational Unit, Owner, and See Also. Under 'See Also', there are two user profiles: Lou Reed and Chris Kidman. Below this is a 'Members To Add' section with a 'Select Member' button. A 'Search for Members to View/Delete' section provides instructions: 'To delete members from the group, check the box(es) next to the member(s) and click the Save button.' At the bottom, there is a 'Select Member Type' section with radio buttons for 'People' (selected) and 'Groups'. A second search bar is located at the very bottom of the page.

4. Select the Member Type you are searching for in this group:

- Select People to search for users. Search results can include static, nested, and dynamic users.
 - Select Groups to search for groups. Search results can include static and dynamic nested groups.
5. From the Search Members By drop-down list, select an attribute as the basis for the search.
 6. Select a search operator.
 7. Enter search criteria.
 8. Click Go.

The Manage Group Members page displays two levels of nested groups and their members in the search results. This includes a child nested group, its members, and its children.

Search for Members to View/Delete
To delete members from the group, check the box(es) next to the member(s) and click the Save button.

Select Member Type Peoples Groups

Search Members by

Search Results:

Static Members

Name
<input type="checkbox"/> Rex Hudler
<input type="checkbox"/> Shawn Delaney

Dynamic Members

Name
Ashely Culley

Nested Members
 No nested members found, with the above search criteria.

Customizing Search Results for Group Members

Note that all search results are subject to access control.

To customize search results

1. Search for group members from the Manage Group Members page. See “Searching for Group Members” on page 147.
2. Click the Customize button on the Manage Group Members page.

3. Select one or more columns you want displayed in the search results and click Save.

Deleting Group Members

You can delete group members displayed in the search results from the Manage Group Members page. You can only delete static members. You cannot delete dynamic or nested members.

To delete group members

1. Search for group members from the Manage Group Members page. See “Searching for Group Members” on page 147.
2. Select the check the box or boxes next to the members you want to delete.
3. Click Save on the Manage Group Members page.

Adding Group Members

You can add members to a group.

To add group members

1. Go to the Manage Group Members page, as described in “Searching for Group Members” on page 147.
2. From the Manage Group Members page, click the Select Members button beside the Members To Add field.

The Selector page appears.

3. From the Selector page:
 - If you want to add users to this group, select the person member type.
 - If you want to add nested groups to this group, select the group member type.
4. Click Add for each member you want to add.
5. Click Done.
6. Click Save on the Manage Group Members page.

Managing Group Subscriptions

The Group Manager provides the ability for users to subscribe and unsubscribe to groups.

Only groups configured as Oblix Advanced Groups can include a subscription policy. The `oblixAdvancedGroup` is provided by NetPoint to give you attributes that you might need when working with groups. Table 17 shows the contents of `oblixAdvancedGroup`:

Table 17 Contents of `oblixAdvancedGroup`

Attribute	Characteristics
<code>obGroupAdministrator</code>	Display Name: Group Administrator Semantic Type: Group Administrator Display Type: Object Selector
<code>obGroupDynamicFilter</code>	Display Name: Dynamic Filter Semantic Type: Group Dynamic Member Display Type: Filter Builder
<code>obGroupExpandedDynamic</code>	Display Name: Group Expansion Semantic Type: None Display Type: Radio Buttons Comment: This attribute is used for expanded dynamic groups.
<code>obGroupPureDynamic</code>	Display Name: Dynamic Members Only Semantic Type: None Display Type: Radio Buttons Comment: This attribute indicates whether the group is purely a dynamic group. It affects subscriptions.
<code>obGroupSimplifiedAccessControl</code>	Display Name: Group Access Semantic Type: None Display Type: Radio Buttons Comment: This attribute is used for creating a group workflow. It controls the simplified access control feature.
<code>obGroupSubscribeMessage</code>	Display Name: Subscription Message Semantic Type: None Display Type: Multi-Line Text Comment: This attribute is used for subscription notification.

Table 17 Contents of oblixAdvancedGroup

Attribute	Characteristics
obGroupSubscribeNotification	Display Name: Notification Semantic Type: None Display Type: Check Box Comment: This attribute is used for subscription notification.
obGroupSubscriptionFilter	Display Name: Subscription Filter Semantic Type: None Display Type: Filter Builder Comment: This attribute is used for group subscriptions using a filter.
obGroupSubscriptionType	Display Name: Subscription Policy Semantic Type: None Display Type: Selection Menu Comment: This attribute is used for group subscriptions.
obGroupUnsubscribeMessage	Display Name: Unsubscription Message Semantic Type: None Display Type: Multi-Line Text

Note: If you create a static group with one or more members and then modify the group so that the Dynamic Members Only flag is set to true, NetPoint allows you to do so without issuing a warning.

Subscribing to Groups

There are three ways a user can subscribe to a group, assuming the Master Identity Administrator configured a group subscription policy for that group:

- From the Group Profile page in Group Manager
This enables users to subscribe to the selected group displayed in the profile.
- As the last step of a Create User workflow
Users can subscribe to multiple groups during the last step of a create user workflow. See “Chaining COREid Functions Into Workflows” on page 171 for more information.
- From the Manage Subscriptions page in Group Manager
This enables users to subscribe to multiple groups from the Manage Subscriptions page.

To subscribe to multiple groups

1. From the Manage Group Members page, click the Manage Subscriptions function.
2. Select the check box next to each group you want to subscribe to.
3. Click Save Subscriptions at the bottom of the Manage Subscriptions page.

A list of groups to which you are subscribed appears. This includes:

- All groups with an open subscription policy.
- All groups with filter subscription policy, and you satisfy the filter criteria.
- All groups controlled through a workflow subscription policy where you are a participant in the initiating step of the change-attribute workflow that applies to these groups.

Configuring Logging and Auditing Policies

NetPoint allows you to capture information about user actions performed within each COREid application. Captured information is stored as *audits* and *logs* of COREid events.

To audit and log user activity that is specific to a COREid application, you must configure auditing and logging policies that are specific to the application. These settings do not overlap with the global auditing and logging policies for a COREid Server. Each COREid Server has one logging and one auditing file. Each COREid System application can be configured for logging and auditing.

- The default logging file is: *COREidInstall_dir/oblix/logs/logfile.lst*.
- The default audit file is: *COREidInstall_dir/oblix/engine/auditfile.lst*

For information about changing the default file names, see “Managing COREid Servers” on page 284.

Viewing Logging and Auditing Policies

You can view logging and auditing policies from each COREid application.

To view logging and auditing policies

1. In the COREid System Console, click User, Group, or Organization Manager Configuration > Configure Logging and Auditing Policies.

The Configure Application Logging Policy/Configure Application Auditing Policy page appears, displaying the following information:

Item	Description
Logging	
Application Log Level	Level of NetPoint events being logged
Auditing	
Profile Attributes	Attributes that describe the profile of the user triggering the event
Event Name	NetPoint operation being audited
Application Auditing Enabled	Indicates whether or not auditing is enabled for this event
Audit Success	Indicates whether or not event successes are audited
Audit Failure	Indicates whether or not event failures are audited

Modifying Logging and Auditing Policies

If you have appropriate permissions, you can change any logging or auditing policy that you can view. These settings do not overlap with the Configure General Logging and Auditing Policies feature found under COREid Configuration > *Application Name* > Common Configuration.

To set or modify logging and auditing policies

1. In the COREid System Console, click User, Group, or Organization Manager Configuration > Configure Logging and Auditing Policies.
2. Click Modify in the lower part of the page.

The Modify Application Logging Policy and Modify Application Auditing Policy page appears.

3. In the Application Log Level field, select the events to be logged:

Event	Description
Debug	Logs all messages. However, when you choose Debug in the Manage Logs page, only Debug messages are displayed.
Info	Logs all messages.
Warning	Logs both warning and error messages.
Error	Logs only error messages.
No logging	No events are logged.

Note: The more general the category of message, the more messages are logged. For example, more messages are logged if you pick Warning instead of Error, and even more if you pick Info instead of Warning. A growing number of messages consumes more disk space and can impact performance.

- In the Profile Attributes fields, select the attributes that can trigger events you want to audit.
- In the Application Auditing Enabled column, select each event you want to enable for auditing.
- In the Audit Success and Audit Failure columns, select each event you want to audit.

For example, you can audit every Modify Location event, but audit only View Profile *failures*.

EventName	Application Auditing Enabled	AuditSuccess	AuditFailure
Search	yes	no	yes
View Profile	yes	no	yes
Modify Profile	yes	yes	yes
View Location	yes	no	yes
Modify Location	yes	yes	yes
Substitute Right	yes	yes	yes
Workflow	yes	yes	yes
Configuration	yes	yes	yes
Deactivated User	yes	yes	yes

Modify

- Click Save.

You return to the previous page.

Generating Reports

Reports enable you to view information about an object class. Reports provide an alternative to searches and enable you to report on attributes that are not available from a search.

Configuring Reports

Master Identity Administrators must define a report from the COREid System Console before users can view the report in the User Manager application.

For example, after configuring an Employees tab for User Manager, as described in “Viewing and Modifying Tab Configuration Information” on page 101, you can create reports listing employees in a specific building, employees with specific job titles, or employees in a particular department.

To configure a report

1. From COREid System Console, click User Manager Configuration > Configure Reports > List Reports.

The following page appears the first time you create a report.



2. Click Add to display the Query Builder on the Configure Reports page.
3. Select the first Attribute for the basis of your report criteria, then click Add.
4. From the list beside the attribute, select the appropriate method.
5. Enter the report criteria.

The format of this criteria depends on the attribute display type.

6. Repeat steps 3-6 for any additional attributes you want added to this report.

Note: When you select more than one attribute for a report, you must select whether this is an And or an Or operation. See the sample page below.

7. Click Test to verify that the report generates data correctly.

A page similar to the following one appears.

Obliv • NetPoint COREid System Console

System Admin | User Manager Configuration | Group Manager Configuration | Org. Manager Configuration | Common Configuration

Configure Tab

Configure Reports

Configure Auditing Policies

Configure Reports

You can create, modify, publish and delete reports for each User Manager tab. Select a tab and click List Reports to display currently configured reports for that tab.

List Reports For:

Obliv • NetPoint Query Builder

Basic | Advanced

Select Attribute:

Phone Number

And

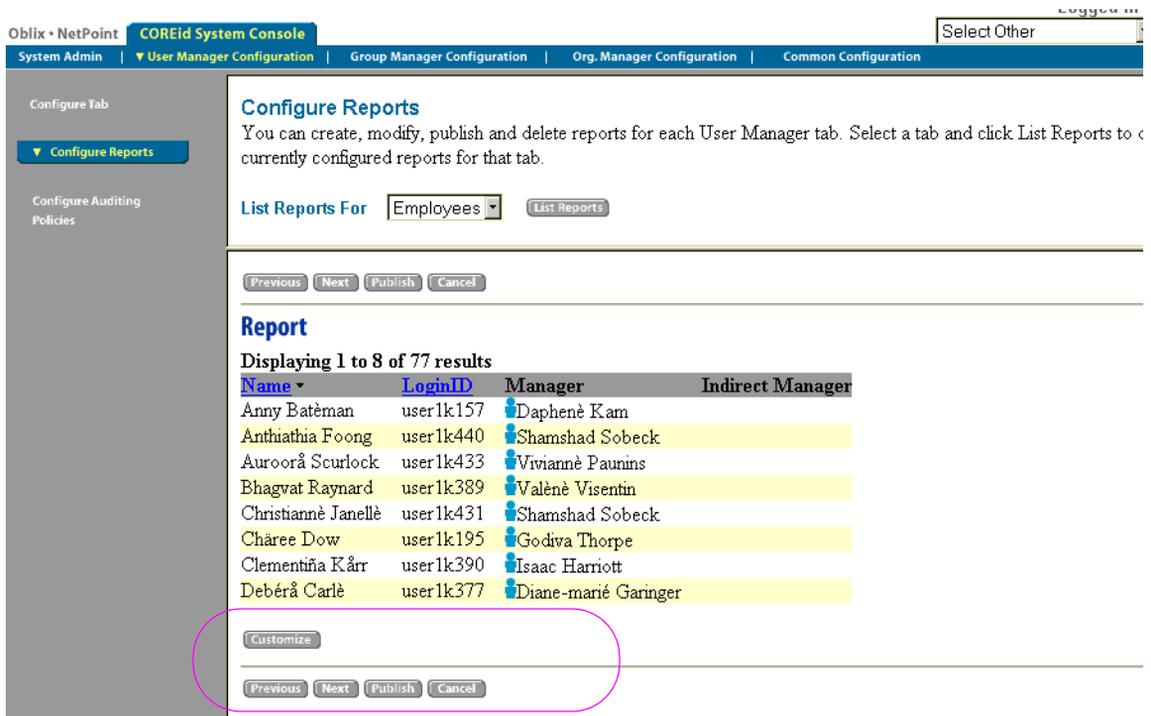
All 8 results

Test Results

Name	LoginID	Manager	Indirect Manager
Anny Batèman	user1k157	Daphenè Kam	
Anthiathia Foong	user1k440	Shamshad Sobeck	
Auroora Scurlock	user1k433	Viviannè Paunins	
Bhaqvrat Raynard	user1k389	Valènè Visentin	

8. Click Save.

A page like the following one appears. Several buttons become available and are highlighted in the screen below. These will be used in the next procedure.



To format and publish the report

1. From the Configure Reports page, click Customize to customize the report column headings.
2. Customize the column names in the form that appears, then click Save.
3. Click the Publish button.
4. Enter a Name and an optional description for this report.
5. Click Save to make this report available in the User Manager application, under the Reports tab.

Viewing, Modifying, Localizing, and Deleting Reports

Viewing reports is subject to access control and searchbase settings.

You can display a report's name and description in more than one language if you install the appropriate language packs and configure them for those languages. See "Configuring Multiple Languages for NetPoint" on page 272 for more information.

To view or modify reports

1. From COREid System Console, click User Manager Configuration > Configure Reports.
2. Select the type of report you want to view or modify from the drop-down list.
3. Click List Reports.
4. Select the link to the report you wish to view.
5. Click the Customize button to change the report criteria.
6. Click Save to save the new report format.

See “Configuring Reports” on page 156 for more information on publishing reports for others to view.

To localize reports

1. From COREid System Console, click User Manager Configuration > Configure Reports.

2. Click List Reports.

All existing reports are listed on the page.

3. Click the report that you want to localize.

The report details appear on the page.

4. Click Publish.

The Publish Report page appears. This page contains the links for all the installed languages.

5. Click the language in which you want to publish the report.

6. In the Report Name field, enter a display name in the selected language.

7. In the Report Description field, enter a brief description of the report.

This information is optional.

8. Click Save to save your changes (or Cancel to exit the page without saving your changes).

The reports are displayed in the User Manager.

To delete reports

1. From COREid System Console, click User Manager Configuration > Configure Reports.

2. Select the tab that contains the report you want to delete.

3. Click List Reports.

4. Select the (-) icon next to the report name to delete it.

Advanced Configuration

The following sections describe expanding dynamic groups, limiting the scope of a directory search, and editing an XML file to configure attribute permissions.

Expanding Dynamic Groups

If a group's membership is determined by an LDAP filter, you can generate a static membership list by expanding the group. Generating a static list saves NetPoint from having to run the LDAP filter with every group access.

Group expansion updates the static list by running the LDAP rule that specifies dynamic membership, then storing the results in the static member attribute. Many NetPoint functions test a group for membership. Since testing static membership is faster than testing dynamic membership, it is preferable to find a member in a static list. Also, third-party applications may only be able to check static membership. Frequent expansion keeps static membership accurate for third-party applications.

The group expansion operation itself is an expensive process. However, you can expand a group as a background process so the impact is hidden from users.

Note: If you have static members in a dynamic group and you expand the group, the original list of static members is overwritten with the members who currently satisfy the filter criteria. This is true even if you have set the flag for dynamic members only to false. The filter overrides other group settings.

Before a user can expand a group, two conditions must be met:

- The `obgroupexpandeddynamic` attribute must be set to true.
- The person expanding the group must have Read permission for two attributes, `obgroupexpandeddynamic` and `obgroupdynamicfilter`. The user also must have Write permission for the attribute assigned the Group Static Member semantic type.

See the table in “Managing Group Subscriptions” on page 151 for a breakdown of NetPoint-supplied group attributes.

To expand a dynamic group

1. In Group Manager, click the Configuration option at the top of the page.
The Configuration page appears.
2. Click Expand Dynamic Groups.
The Expand Dynamic Group page appears.
3. Select one of these options:

- Select By Group and click Select Group to choose one or more groups
 - Select All to expand all groups
4. Click Expand.
The Expanded Groups page displays a list of all groups that have been expanded.
 5. Click the group link to display the Group Profile page for that group.
 6. Click Done.

Modifying the Default Searchbase Scope

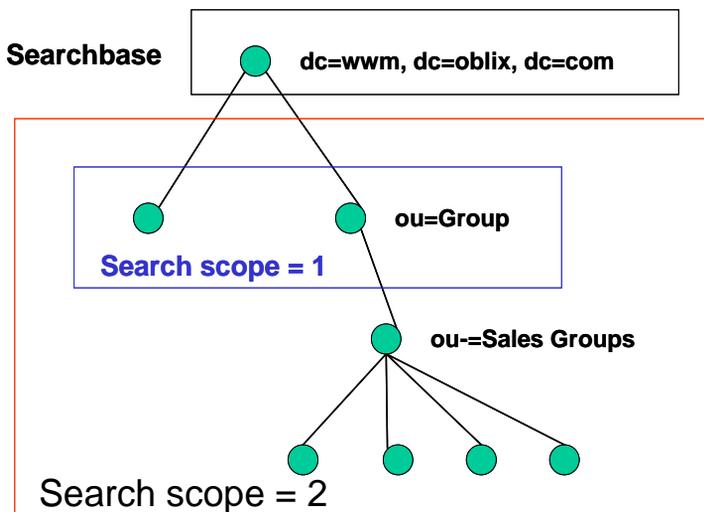
Some portions of NetPoint call out to external XML files to get configuration information. The GlobalParams.xml file is one such file. This file controls search scope among other things.

By default, the search scope is set to subtree for the COREid System, meaning that the search begins at the starting point of the searchbase and includes its children. Depending on the size of your directory, you may want to change the default search scope using the ResourceFilterSearchScope parameter. The possible values for this parameter are:

- **1**—Search only one level directly below the top node of the searchbase.
- **2**—Start at the top node of the searchbase and proceed to the bottom node.

Figure 6 shows that setting ResourceFilterSearchScope to 1 could limit results to just a few returned entries, while setting it to 2 could return thousands of entries.

Figure 6 Search-Scope Options



To set the globalParams.xml file

1. Locate the globalParams.xml file in the following directory:
COREid_install_dir/identity/oblix/apps/common/bin
2. Back up the file.
3. Open the file in an ASCII editor (for instance, Notepad) or an XML editor.
4. Find the ResourceFilterSearchScope parameter and change the value.
5. Restart WebPass and the COREid Server.

Simplified Attribute Permissions for a Group

Simplified attribute permissions lets a group creator select Read, Write, and Notify permissions without having to set permissions for each attribute as described under “Setting and Modifying LDAP Attribute Permissions” on page 138.

Simplified permissions are applied to newly created groups where the management domain of the policies is the DN of the new group. Later, these policies can be modified through the access control feature.

Implementing Simplified Permissions

An administrator can configure as many sets of simplified permissions as needed. The administrator creates permissions in the *COREidInstall_dir/oblix/apps/groupservcenter/bin/gscacparams.xml* file.

This file contains embedded compound lists to define the roles, users, and groups the model applies to, the rights assigned, and the attributes to which the rights apply. When this file is applied to a new group, an access control entry is created for each right in the file.

Sample gscacparams.xml File

The following is a sample set of permissions within a gscacparams.xml file. The model name is Public:

- In entry 1, the role is ob_any, the right is read, and the attributes are description, uniquemember, and owner.
- In entry 2, the role is owner, the right is write, and the attributes are description, uniquemember, and owner.

```

<?xml version="1.0"?>
<ParamsCtrlg xmlns="http(s)://www.oblix.com" CtrlgName="gscaclparams">
<!--#----->
<!-- #Access Control Functions -->
<!--#----->
<!--#----->
<!-- # Public access -->
<!--#----->
<CompoundList ListName="">
<CompoundList ListName="Public">
<CompoundList ListName="entry1">
<ValList ListName="roles" >
    <ValListMember Value="ob_any">
</ValList>
<ValList ListName="rights" >
    <ValListMember Value="READ" Operation="Add"/>
</ValList>
<ValList ListName="attributes" >
    <ValListMember Value="description"/>
    <ValListMember Value="cn"/>
    <ValListMember Value="uniquemember"/>
    <ValListMember Value="owner"/>
</ValList>
</CompoundList>
<CompoundList ListName="entry2">
<ValList ListName="roles" >
    <ValListMember Value="owner" Operation="Add"/>
</ValList>
<ValList ListName="rights" >
    <ValListMember Value="WRITE" Operation="Add"/>
</ValList>
<ValList ListName="attributes" >
    <ValListMember Value="description" Operation="Add"/>
    <ValListMember Value="cn" Operation="Add"/>
    <ValListMember Value="uniquemember" Operation="Add"/>
    <ValListMember Value="owner" Operation="Add"/>
</ValList>
    </CompoundList>

```

Simplified Permissions Reserved Words

The following table summarizes the reserved words for simplified permissions.

Reserved Word	When Used	Description
rights	Once per entry	Specifies the right: read, modify, or notify.
attributes	Once per entry	List that specifies the attributes. Any group object attribute can be added to the list.
roles	Once per entry	Roles to which entry applies. Roles can be any pre-defined role, such as uniquemember, owner, ob_any, or ob_anonymous.
people	Once per entry	Specifies the distinguished names to which this entry applies.
source	Once per model	Specifies the base uid of the users who will see this model. If a base uid is not specified, everyone can see this entry.
target	Once per model	Specifies the base uid of the target where this model applies. If the group is not part of this base, the rights cannot be set.

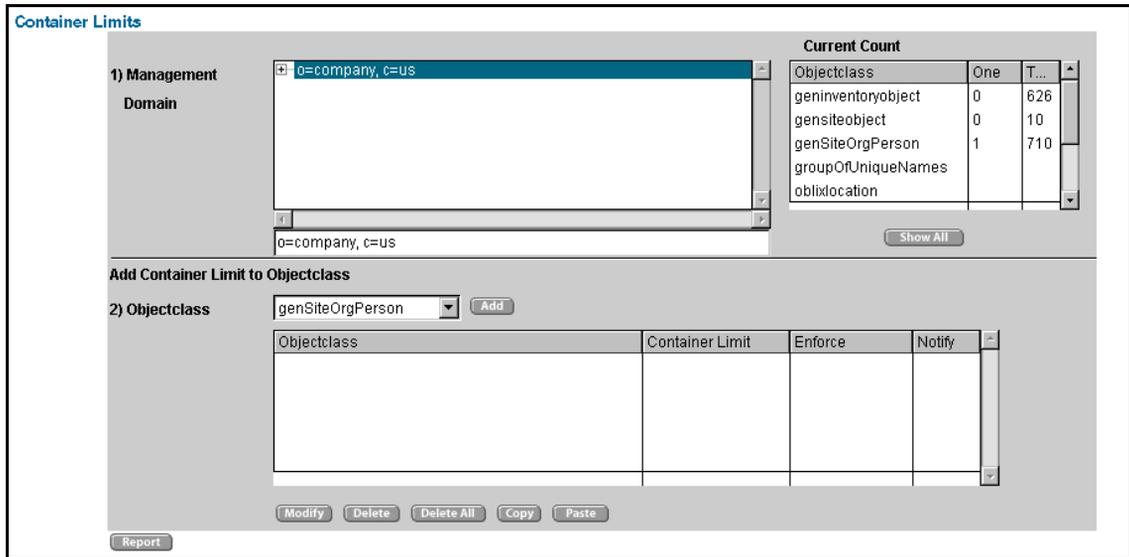
Setting Container Limits in Organization Manager

Use the Container Limits function to control the number of objects and child objects for an organizational unit and its object classes. You can define who receives notifications when the limit is about to be exceeded. For example, you can have organizational units in your directory tree that you use for storing extranet customers. You can limit to 10,000 the number of customers with access to your extranet portal.

Note: The Container Limits feature counts the number of objects from the directory. If the number of objects is very large, performance can be affected.

To view and add container limits

1. From the Organization Manager, click the Configuration > Container Limits.
The Container Limits screen appears.



In the example above, the gensiteOrgPerson object class has 1 child stored at the current level of the DIT and 710 total children at this level and below. See the Current Count table, the column labeled:

- **One**—The number of direct children in each object class for the selected directory information tree (DIT) entry.
- **Total**—The total number of children in each object class for that DIT entry.

2. In the Management Domain box, select a DIT entry you want to view.

The Current Count box displays all configured structural classes associated with the entry and the number of their children.

The Objectclass table displays the container limit, enforcement, and notification policies for the selected DIT entry, listed according to object class.

3. Select an object class and click Add to add a container limit, in the Objectclass drop-down list.

A second Container Limits screen appears showing the Management Domain and Object class you selected in the previous screen.

4. In the Container Limit box, specify the maximum number of children this object class can contain at this DIT level.
5. When you want to notify someone by email that your object class is nearing its container limit, select Notify if used up, and specify the limit percentage when you want the email sent.
6. Select Override subordinate policies to create a container limit that cannot be overridden by a lower policy on the DIT.
7. Use one or more of the following to specify the persons to receive container limit warnings:
 - Select Build Filter, then use the Query Builder to create a rule.
 - Click Select User, then use the Selector to specify one or more users.
 - Click Select Group, then use the Selector to specify one or more groups.

The Users, Roles, and Rules fields have an *or* relationship. Users specified in any of the fields are notified.
8. Click Save to save your container limit and add it to the Objectclass table.

Copying Container Limits

You can copy container limits from one domain to another.

To copy container limits from one domain to another

1. From the Organization Manager, click Configuration > Container Limits.

The Container Limit screen appears (as shown on

2. In the Management Domain box, select the directory information tree (DIT) entry you want to view.

The Current Count box displays the structural classes associated with the entry and the number of their children.

The table Add Container Limit to Objectclass displays the container limit, enforcement, and notification policies for the currently selected DIT entry, listed according to object class.

3. Click Copy.
4. In the Management Domain box, locate the destination entry where you want to add the container limits.
5. Click Paste.

The container limit policies are added to the selected DIT entry.

Modifying Container Limits

You can change container limits. See “Setting Container Limits in Organization Manager” on page 165.

To modify a container limit

1. In the Organization Manager, click Configuration.

The Configuration screen appears.

2. Click Container Limits.

The Container Limits screen appears.

3. In the Management Domain box, select the DIT entry you want to view.

The Current Count box displays all configured structural classes associated with the entry and the number of their children.

4. In the Add Container Limit to Objectclass panel, select an object class from the Objectclass column.
5. Click Modify.

The second Container Limits screen appears.

6. Make your changes.
See “Setting Container Limits in Organization Manager” on page 165 for information about these fields.
7. Click Save.

Deleting Container Limits

You can delete a container limit.

To delete a container limit

1. In the Organization Manager, click Configuration > Container Limits.
2. In the Management Domain box, select a directory information tree (DIT) entry.

The Current Count box displays all configured structural classes associated with the entry and the number of their children.

3. In the Add Container Limit to Objectclass panel, select an object class.
4. Click Delete.

The object class container limit is deleted.

Note: Click Delete All to delete all container limits for a DIT entry.

5 Chaining COREid Functions Into Workflows

This chapter includes the following topics:

- “About Workflows” on page 171
- “Using the QuickStart Tool” on page 190
- “Using the Workflow Applet” on page 194
- “Advanced Workflow Ticket Routing” on page 212
- “Defining a Subflow” on page 210
- “Performing Asynchronous Operations” on page 228
- “Using a Workflow” on page 229
- “Managing Workflows” on page 235
- “Advanced Workflow Options” on page 244
- “Creating a Self-Registration Workflow” on page 247
- “Creating a Location Workflow” on page 250

About Workflows

A COREid *workflow* enables Master Identity Administrators and Delegated Identity Administrators to apply external business logic to COREid functions, thereby organizing and automating complex procedures such as the creation of benefits and email accounts for new employees or the modification of user profile attributes in the NetPoint directory.

Each workflow consists of a sequenced chain of actions. Rather than making a single person responsible for completing all the tasks in the workflow, you can assign each step to the specialist most appropriate to perform that step. When a step is completed, the workflow engine can send the workflow ticket to the person responsible for the next step in the sequence.

In sum, workflows enable you to:

- Automate and standardize processes for creating objects, deleting objects, and modifying attributes in the NetPoint directory.
- Apply data integrity and rule checking when creating objects, deleting objects, and modifying attributes.
- Configure COREid as a data entry system for provisioning back-end applications.

Typical Workflow Examples

Workflows are appropriate for just about any frequently repeated, multi-step task involving any combination of user actions or automated data retrieval. Each workflow is associated with one of the COREid applications. The following list covers some common workflows:

- **User Manager**—You can define a workflow to permit users to change their department number and phone number pending approval by a manager. You can ensure that when a new user is created, the appropriate people obtain information about this person programmatically from an external system.

A different workflow can add new users to your corporate email application. If you have defined an object template schema, you can use a workflow to send data from a COREid application to a back-end application for provisioning. See “Provisioning External Applications from COREid” on page 253 for details on object templates. Oblix also provides a solution that uses a workflow to provision applications managed by MIIS. See the *NetPoint Integration Guide* for details.

- **Group Manager**—You can create a workflow to route group registration requests to a manager for approval.
- **Organization Manager**—You can give a supplier the ability to create entries for parts, pending manager approval of each entry the supplier adds. You can also create a workflow that first enables a user to add a new part entry, then routes the request to add the data to an appropriate person for approval, and finally, permits the person giving approval to commit the new data to the directory.

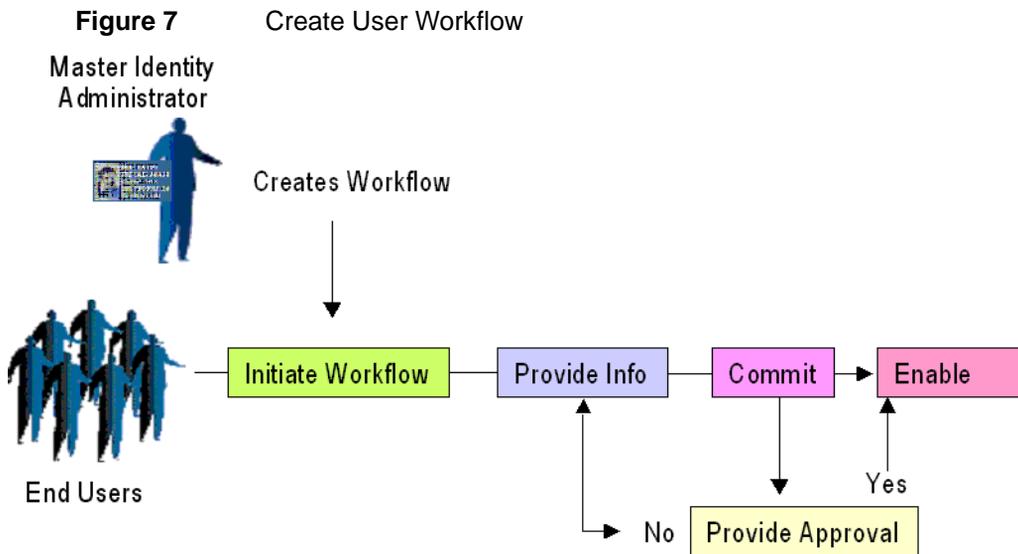
Advanced Workflow Options

NetPoint workflows support the following advanced features:

- *Subflows* enable certain workflow activities to occur in parallel. For instance, if a request to create a new user requires approvals from two different departments, both parties can receive approval requests simultaneously. For details, see “Defining a Subflow” on page 210.
- You can route specific workflow steps to different *dynamic participants*, who are selected on the basis of attribute values or business logic evaluated at runtime. For details, see “Specifying Dynamic Participants” on page 214.
- You can designate *surrogates* to assume responsibility for a step when the primary participant assigned to that task is out of the office or otherwise unavailable to process incoming tickets. For details, see “Specifying Surrogates” on page 221.
- You can configure *time-based escalation* so that a workflow ticket is routed to a different participant if the original participant does not complete the assigned step within a specific period. For details, see “Enabling Time-based Escalation” on page 224.
- You can invoke a workflow from any Web page as a *portal insert* or application using IdentityXML. See the *NetPoint 7.0 Developer Guide* for information on IdentityXML.
- Workflow *auditing* enables you to monitor the state of a workflow and to determine exactly who performed particular actions at each step in the process. “Monitoring a Workflow” on page 233.
- For actions that do not require human intervention, you can configure workflow steps so that COREid automatically obtains the required data from external sources. See the *NetPoint 7.0 Developer Guide* for information on IdentityXML.

Workflow Types

Workflows come in various *types*. For instance, one type of workflow allows you to change one or more attributes for an existing object. Another type of workflow allows you to create a new object. Figure 7 illustrates a Create User workflow:



Creating Workflows

The following overview summarizes the high-level steps for creating a workflow. The actual steps vary slightly for Change Attribute workflows and Create User (or Group, or Object) workflows:

Task overview: Creating a workflow definition

1. Add objects to the tab for the relevant COREid application.
2. Configure attributes for the objects.
3. Configure read and write permissions for LDAP attributes.

Participants in a workflow must have appropriate read and write permissions for LDAP attributes that are viewed and changed during the processing of the workflow. See “Allowing Users to View and Change LDAP Data” on page 126 for details.

4. Add the attributes to the application panels.

This applies to both LDAP and template attributes. For Change Attribute workflows, users will not see the attributes from template objects on the profile page that contains the panel. However, the template attributes must be added for the workflow to operate properly.

5. Configure the workflow.

As discussed in “Defining Step Attributes” on page 203 for details, when you add attributes for a step in a Change Attribute workflow, the topmost attribute in the list must have been configured on a profile page. The subsequent attributes in the list will be added to the page automatically as long as the topmost attribute has been configured correctly.

A workflow definition changes the appearance of its related profile page in the COREid application for which the workflow was created. For example, when a Modify Attribute workflow has been configured correctly, a “modify” button appears next to the attribute on the Modify Profile page for the target object. When a Create User workflow has been configured correctly, the desired attributes appear when you select Create User Identity in the User Manager.

How Users Access Workflows in a COREid Application

After a workflow definition has been created, an instance of the workflow is initiated in one of several ways, depending on the type of workflow that is being used:

Table 18. Methods for initiating Workflows

Workflow Type	User Initiates This Workflow From. . .
Change Attribute	A Request to Modify button on a Modify Profile page for the user
Create User	The Create User Identity page that is accessed from the Create User Identity link in the User Manager.
Deactivate User	An Initiate User Deactivation button on the View Profile page for the user.
Reactivate User	An Initiate User Reactivation button appears on the View Profile page when you have created a Reactive User workflow. You first must find the user from the Deactivated User Identity page in the User Manager. A Reactivate user operation can be done by a Directory Administrator or a user with reactivate privileges.
Self-Registration	When this type of workflow is created, a URL is generated that initiates this workflow. You must save the URL and use it to initiate the Self-Registration workflow.
Create Group	The Create Group page that is accessed from the Create Group link in the Group Manager.
Delete Group	The View Profile page for the group.
Create Object	Create page in the Organization Manager.
Delete Object	View Profile page for the object.

About Workflow Tickets

As program execution reaches a given step in a workflow, the workflow engine creates a *ticket* for that step instance. During a Create User workflow, for example, a ticket is typically sent to specific participants in IT as soon as the user selects the Create User function in the User Manager.

Each workflow ticket is initially displayed in the form of a link. When the participant clicks this link, he or she is prompted to perform the action associated with that step in the workflow. For example, when someone in IT processes a ticket for a Create User workflow, he or she is typically prompted to supply a login id and password for the new user.

A workflow log is created upon completion of each step in the workflow.

For more information, see “Using a Workflow” on page 229.

The following screen shot illustrates a Create User Identity page. The contents of this page are based on attributes configured in a Create User workflow:

Create User Identity
Select Template

Create user - Advanced (with approval) ▾

Full Name*

Last Name*

Login

Mail

Mobile

Out Of Office Indicator True False

Room Number

Street

Telephone Number

User Password

New User Password

Retype User Password

Comment

Save Cancel

Once information about a new user is saved on this page, the “initiate” step of the workflow is complete. The workflow definition generates a ticket for this workflow instance, as illustrated below:

Confirmation

Initiate - Completed

Ticket generated for next step
Approval [77b72fdf7d384b59b7e289a1a6832a55_2](#)

A participant in this workflow can view the ticket generated for this workflow step, and can approve the addition of the new user. Ticket information, which you display by clicking on the ticket number next to the approval label in the preceding graphic, is illustrated below:

Ticket Information

Ticket Number	77b72fdf7d384b59b7e289a1a6832a55.2
Request Type	Create User
Action	Approval
Status	 Pending for Participant
Application Name	User Manager
Date Created	08/26/2004
Service	Full Name Last Name Login Mail Mobile Out Of Office Indicator Room Number Street Telephone Number User Password
Requested By	admin admin
Requested For	Jill Smith
Participants	View
Retry Count	
Locked By	
Workflow Name	Create user - Advanced (with approval)
Parent Request Number	

Pending Requests

Ticket Number	Application Name	Action	Status
77b72fdf7d384b59b7e289a1a6832a55.2	User Manager	Approval	 Pending for Participant

A Workflow Scenario

Suppose you create a workflow for adding a user in NetPoint. You could define a Create User workflow that performs the following steps:

Process overview: Creating and using a Create User workflow

1. From the User Manager application, you create a new workflow definition.

In this example, the workflow definition has three steps and specifies that anyone in IT who has logged in to the User Manager can create a new user workflow:

Step 1: Initiate—This step allows anyone who has logged in to the User Manager to input data for a new user.

Step 2: Provide Information and Approval—This step allows a person in HR to approve the data entered for the user.

Step 3: Activate—This step activates the new user.

2. A user logs in to the User Manager.
3. The user selects a Create User button.
The workflow instance prompts the user to supply a name, user ID, and password for the new user, plus the userid and email of the new user's manager.
4. The workflow instance then routes a request to create the new user, along with information about the new user, to the manager of that user.
5. The manager clicks the Requests function in the User Manager application to display the request in the form of a link to a job ticket.
6. The manager clicks the link for the ticket to display the request.
7. To approve the request, the manager clicks a Process Request button.
8. In the Process Requests page, the manager clicks an Approve button.
9. NetPoint processes the request and the new user is enabled in NetPoint.

The user is now allowed to log in and use the functions they are entitled to as defined by their directory profile and the rights assigned to attributes in that profile by a NetPoint administrator. See “Allowing Users to View and Change LDAP Data” on page 126 for details.

LDAP Versus Template Attributes in a Workflow

When you define a workflow, you have a choice of using two types of objects and attributes in most workflow steps:

- **LDAP Objects and Attributes**—You can use a workflow to modify objects and attributes that you have configured for an application profile page. The people who participate in the workflow must have appropriate privileges for viewing and modifying these objects and attributes.
- **Template Attributes**—If you are using a workflow to provision a back-end application, you configure workflow steps for adding information based on a template schema. When template attribute values are committed during the workflow, an Identity Event API plug-in can intercept this data and send it to a back-end application for provisioning. See “Provisioning External Applications from COREid” on page 253 and the *NetPoint 7.0 Developers Guide* for details.

In NetPoint 7.0, provisioning allows only for a one-way flow of data from COREid to the back-end system. As a result, you might want to configure provisioning workflows to write data to both the LDAP directory and to the back-end system. This enables your users to view the data that has been configured for the workflow target. However, to see the current state of the target in the back-end application, you must access the application or its logs.

For provisioning workflows, you should have separate Commit, Activate, Enable, Delete, Disable, and Deactivate steps for each schema to which the workflow is written.

Workflow Types, Steps, and Actions

A workflow *type* determines the purpose of the workflow, for example, creating a user. A workflow *step* is a discreet segment of the workflow. Steps are performed in a series. A workflow *action* is an activity performed during a step, such as issuing a request for information.

For example, the Create User workflow type enables you to create a directory entry for a user. This type of workflow can have actions for requesting information about the user, actions for collecting the information, actions for approving the request, and so on.

The following table correlates the different types of workflows to the COREid applications:

Table 19 Workflow Types

Application	Workflow Type and Description
User Manager	<ul style="list-style-type: none"> • Create User—Adds a user to the directory. • Self-Registration—Enables users to add themselves to the directory. • Deactivate User—Makes a user unable to log in and unavailable for viewing in the COREid System. Deactivation takes effect once a user has logged out. It removes a user's future access to the system. An administrator with sufficient access privileges can view deactivated users and either permanently delete them or reactivate them. • Reactivate User—Displays the Initiate User Reactivation button on the User Profile page and changes the status of a deactivated user, allowing the user to log in to and use the COREid System again. • Change Attribute—Changes an attribute value on a user profile. Attributes designated on this workflow will have a Request to Modify button on the target profile page.
Group Manager	<ul style="list-style-type: none"> • Create Group—Adds a group to the directory. • Delete Group—Deletes a group from the directory. • Change Attribute—Changes an attribute value on a group profile. Attributes designated on this workflow will have a Request to Modify button on the target profile page.

Table 19 Workflow Types

Application	Workflow Type and Description
Organization Manager	<ul style="list-style-type: none">• Create Object—Adds an object to the directory.• Delete Object—Deletes an object from the directory.• Change Attribute—Changes an attribute value on an object profile. Attributes designated on this workflow will have a Request to Modify button on the target profile page.• Self-Registration—Enables users to add organization objects to the directory.

About Workflow Steps

You must define at least two steps for each workflow—one to initiate an instance of the workflow and one to finish it. A step consists of the following:

- **A Number**—A unique identifier for this step.
- **Actions**—An action is an activity can occur in the COREid System or in an outside system. Examples include starting the workflow, providing information, and requesting approval. See “About Step Actions” on page 182 for details.
- **Attributes**—An attribute value may be added or modified as part of a step.

For example, you might define a step for changing the value of a user’s phone number attribute. Step attributes may be required, optional, or supplied by completion of another workflow step.

For values that are used locally within the COREid System, you configure LDAP attributes as part of the workflow. For provisioning to a back-end application, you configure both LDAP and template attributes in a workflow step.

Note: If Location ID has the Semantic type DN Prefix it is important to note Active Directory and ADAM do not allow multi-valued RDNs (although iPlanet/SunOne do). For Active Directory and ADAM, ensure that the Attribute Value(s) selection is Single in the meta-attribute configuration.

- **Participants**—A user or users who perform an action.
For example, for a Create User workflow, you may create an Initiate step and configure this step so that anyone who is logged in to the User Manager can start the process for creating a new user. Or you may define a specific participant in a workflow who is responsible for approving a change request. Participants can be assigned based on their role, name, group membership, or another characteristic.

For LDAP attributes, you can also define an LDAP filter that selects participants according to their DN.

- **Target**—The person, group, or other LDAP object that is being created, deleted, and so on.

The target in the workflow definition is an LDAP object, not a template object.

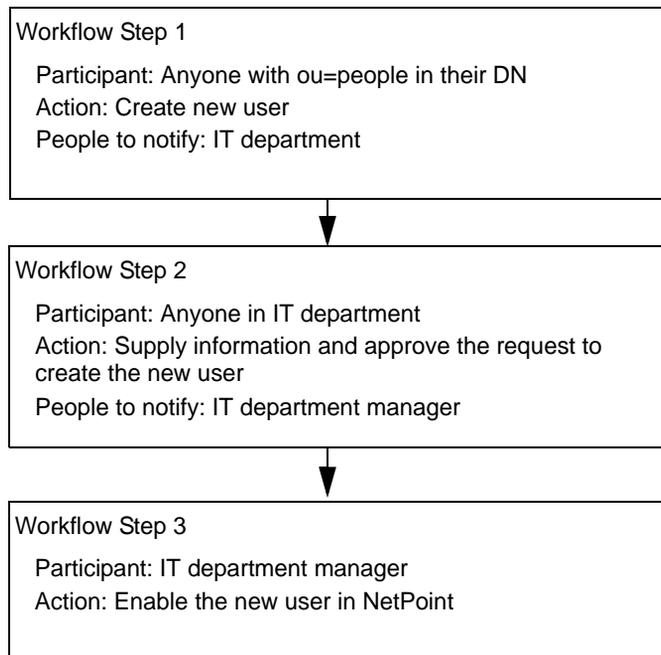
- **Entry Conditions**—A step or subflow that must be completed before the present step.

For example, the first step in a workflow may be the Initiate step. The second step in the workflow may have an entry condition of successful completion of the Initiate step. A typical entry condition is successful completion of the previous step.

- **Notifications**—Users who receive email notification before or after the execution of the step. Other participants can see pending tickets in their incoming request queues whether or not email notification is configured. See also, “Descriptions of Step Actions” on page 186.
- **Pre and Post Processing**—External functions that are executed as part of the workflow. For example, in a create user workflow you might want to have a Java program that is called after an initiating step to assign a unique login id.

A workflow process illustration is shown in Figure 8

Figure 8 Sample Workflow Process



About Step Actions

You assign one action to each step in a workflow. Actions are performed by people or by an automated method.

For example, required actions in a workflow to create a user include:

- Initiating the request.
- Enabling or activating the user.

Available actions depend on the workflow type and the action defined in the previous step. For example, the Initiate action is available for only the first step of a workflow.

Table 20 lists the actions that you can associate with steps in User Manager workflows.

Table 20 Actions Permitted in User Manager Workflows

Workflow Type	Actions
Change Attribute	Request (required) Provide Information Approval Provide Information and Approval Subflow Approval Commit (required) External Action Error Report
Create User	Initiate or Self Registration (one of these two is required) Provide Information Provide Information and Approval Approval Subflow Approval Commit Enable or Activate (one of these two is required) Select Groups Delete Error Report External Action

Table 20 Actions Permitted in User Manager Workflows

Delete User	Initiate (required) Change Information Disable or Deactivate (one of these two is required) Approval Subflow Approval Change Approval Commit Delete Error Report External Action
Deactivate	Initiate Change Information Approval Change Information and Approval Commit External Action Error Report Deactivate Disable Delete
Reactivate	Initiate Provide Information Approval Provide Information and Approval Subflow Approval Commit External Action Error Report Activate Enable

Table 21 lists the actions available in Group Manager workflows:.

Table 21 Actions Permitted in Group Manager Workflows

Workflow Type	Actions
Change Attribute	Request (required) Provide Information Approval Provide Information and Approval Subflow Approval External Action Commit (required) Error Report

Table 21 Actions Permitted in Group Manager Workflows

Create Group	Initiate (required) Provide Information Provide Information and Approval Approval Commit (required) Subflow Approval Delete External Action Error Report
Delete Group	Initiate (required) Change Information Change Approval Subflow Approval Approval Commit (required) Delete Error Report External Action

Organization Manager workflow actions are described in Table 22:.

Table 22 Actions in Organization Manager Workflows

Workflow Type	Actions
Change Attribute	Request (required) Provide Information Approval Provide Information and Approval Subflow Approval External Action Commit (required) Error Report
Create Object	Initiate (required) Self Registration Provide Information Provide Information and Approval Approval Subflow Approval Commit Delete Error Report External Action

Table 22 Actions in Organization Manager Workflows

Delete Object	Initiate (required) Change Information Approval Change Approval Subflow Approval Commit (required) Delete Error Report External Action
---------------	--

Descriptions of Step Actions

Table 23 describes the actions available in workflows.

Table 23 Workflow Step Actions

Action	Description
Activate	User Manager only. Activates a new user in the COREid System. An activated user is enabled, and can log in and perform operations granted by administrators. The obuseraccountcontrol attribute in the user's entry controls activated/deactivated status. The Activate action requires a participant, such as a manager, to activate the user.
Approval	This action can be configured with required attributes. At run time, the values for the required attributes are presented to the participant for approval. No information can be changed by this action.
Change Information and Approval	Performs the same function as the Provide Info and Approval actions, but used only when deactivating users.
Change Information	Performs the same role as the Provide Info action, but is used only when deactivating users.
Commit	Writes the information collected in the previous steps to the directory. A commit operation writes information to the location of the e object in the directory. For example, during a Create operation, the Commit action adds a new entry to the directory. If the workflow contains additional Commit action, the information is written to the location in the directory that contains the newly created object. A Commit action can be used more than once in a workflow. No user action is required.
Deactivate	<p>User Manager only. Deactivation takes effect once the user's current session has ended. A deactivated user cannot log in. Others cannot find a deactivated user in NetPoint except when searching for deactivated users. Deactivating does not delete the user from the directory. The obuseraccountcontrol attribute in user's entry controls activated/deactivated status. A participant is required for a deactivate step in a workflow.</p> <p>Note: To create an .ldif containing deleted users, use the Deactivate or Disable workflow steps instead of Delete. Go to the Deactivated User Identity page, and use the Archive option. This will delete the users from the directory and create a deactivateduser.ldif in the <code>COREid_Install_Dir\oblix\data\common</code> directory.</p>

Table 23 Workflow Step Actions

Action	Description
Delete	The delete action in a Create User, Group, or Object workflow permanently removes the target entry from the directory. It is possible for a Create workflow to be rejected after a target entry is created. The Delete step cleans up the directory so that new attempts to create the same user can be made.
Disable	User Manager only. Deactivates a user, which means the user cannot be recognized by NetPoint once the user's current session has ended. Deactivation takes effect the next time the user attempts to log in. Deactivating does not delete the object from the directory. This action does not require a participant.
Enable	User Manager only. An Enable action is a combination of a Commit and an Activate action. Automatically activates the new user, who is then recognized by NetPoint after the previous step is completed. This action does not require a person to activate the user.
Error Report	When a background process encounters a processing error, you can configure an error report to send the error to particular users. You can also configure an error report when a step is rejected, for instance during the approval process.
External Action	An action performed outside of NetPoint.
Initiate	Starts the Create and Deactivate workflows. This action can be used once in a workflow. It must be the first action. The self-registration action can also be the initiating action of a workflow. All users see the Create Profile button or Initiate Deactivate option on their pages, regardless of whether they have been defined as a participant for this particular workflow. If a user clicks the button or link to the workflow but they have not been defined as a participant in the workflow, an error message will be displayed.
Provide Information and Approval	Combines the Provide Info and Approval actions into one action.
Provide Information	Collects information from the user. This action is similar to Initiate, but it cannot be the workflow's first action.
Request	A user's request to change, add, or delete an attribute. Participants for this action see the Request to Modify or Request to Remove button on the Modify Profile page.

Table 23 Workflow Step Actions

Action	Description
Self-Registration	Lets users complete and submit a registration form. Other participants approve the request and activate the user. This action must be the first step in a workflow. The self-registration action does not necessarily require other participants to approve and activate the new user.
Select Groups	Enables the workflow participant to subscribe a target user to a group or groups during a create user workflow. The new user has to meet the subscription policy. Available only after an Enable or Activate step.
Subflow Approval	Reports the current status of a subflow that has been invoked from a main workflow step. It does not apply to subflows invoked from other subflows.

Note: Email post-notification for a self-registration step requires two parameters in the `globalparams.xml`, `sendMailFromName` and `sendMailFromEmail`. The values for these parameters are placed in the “mail From” or “senders name” and “mail” or “senders email” parts of the SMTP message respectively.

For self registration, these values are provided through `globalparams.xml` because the target is not yet created. In this case, you need to locate the parameters in the `globalparamams.xml`, then modify the values to suit your environment. For example:

```
<SimpleList>
  <NameValuePair
    ParamName="sendMailFromName"
    Value="SelfRegistration"></NameValuePair>
</SimpleList>
<SimpleList>
  <NameValuePair
    ParamName="sendMailFromEmail"
    Value="SelfRegistration@oblix.com"></NameValuePair>
</SimpleList>
```

If the target user has been created the values for `sendMailFromName` and `sendMailFromEmail` are obtained from the naming attribute and email attribute of logged in user’s profile, respectively.

About Subflows

In a simple workflow, all steps execute sequentially. If one step is in a pending state, the workflow does not progress to the next step. Because workflows often involve different participants, this can delay completing the workflow. To speed up processing of a workflow, you may want to define *subflows* that occur in parallel.

Subflows lets you break down workflows into chunks. A subflow can trigger subflows of its own. You can trigger multiple subflows from a single workflow.

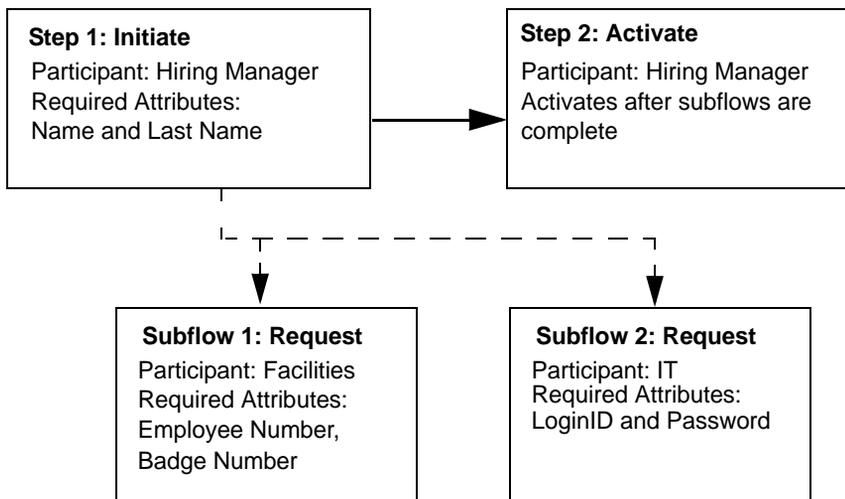
Note: A subflow is always a Change Attribute workflow.

Process overview: A Create User workflow example

1. The hiring manager initiates a Create User workflow.
2. A request is sent to Facilities for an employee number and badge number.
3. A request is sent to IT for the login and password.
4. A request is sent to the hiring manager for final approval and activation of the user.

By using subflows, some of the requests can occur in parallel. The approval waits until the subflows are complete, as illustrated in Figure 9.

Figure 9 Order of step completion when using subflows



A workflow does not move to the next step until a subflow is complete.

Note: For a subflow to launch, the target object or attribute must meet any filter criteria specified in the Workflow Domain filter.

Using the QuickStart Tool

The QuickStart tool enables the rapid creation of simple workflows based on default settings. After completing the QuickStart workflow-definition process, you can use the standard workflow tools to adapt the workflow to specialized needs. For instance, you can specify dynamic participants or surrogates.

The QuickStart tool allows you to define the following workflows:

Table 24 Workflows that Can Be Created with the QuickStart Tool

The workflow named. . .	Contains these steps. . .
Create User, Group or Object (Basic)	Self-Registration or Initiate Commit Error Report Note: For a simple Create User workflow, required attributes are Last Name and Name on most directory servers and Login ID if you use Active Directory.
Create User, Group or Object (Advanced—with Approval)	Initiate Approval Commit Error Report
Self Registration for Users or Objects (Advanced—with Approval)	Self-Registration Approval Commit Error Report
Change Attribute (Basic)	Request Commit Error Report.
Change Attribute (Advanced—with Approval)	Request Approval Commit Error Report

The QuickStart tool assigns anyone who is logged in to NetPoint as the participant for most steps. For the User Manager, the participant in a Change Attribute workflow is any person who has been assigned the role of Manager. For the Group Manager, any person assigned the role of Group Owner is the participant in the Approval step of a Change Attribute workflow.

Once you create a workflow using the QuickStart tool, you can view and modify the workflow steps, participants, affected attributes, and so on. Information on viewing a workflow definition is provided in “Viewing and Exporting a Workflow Summary” on page 235. Information on modifying a workflow is provided in “Modifying a Workflow” on page 237.

Note: Your ability to define a workflow depends on your administrative privileges.

To define a workflow using the QuickStart tool

1. From the COREid System Console, select the User, Group, or Organization Manager.
2. Click Configuration > Workflow Definition.

By default, only NetPoint Administrators, Master Identity Administrators, and Delegated Identity Administrators have permission to view configuration information.

3. Click the QuickStart link (the “Click here” link in the illustration below).

Workflow Definition

Workflow Definition allows an administrator to define different workflows for different organizations. Different workflows pertain to different steps, attributes, and participants. To create a workflow using Quickstart [Click here](#)

[1 of 3] Workflow Definition



This launches the QuickStart tool.

4. Select the type of workflow you want to create.

Note: You can define a Create workflow and a Change Attribute workflow from the same QuickStart page. Scroll to the bottom of the page to see the Change Attribute fields and options.

You can also provide a name for your workflow. A default name is provided, but it does not change if you use the QuickStart tool to create multiple workflows of this type.

Workflow Definition - QuickStart (1 of 2)

Welcome to Workflow Definition - QuickStart. This feature will allow you to create different types of 'Create' workflows, as well as 'Change attribute' workflows.

NOTE: The participants for these workflows will be set to 'Anyone'. This means that any authenticated user will be able to initiate these workflows. If this is not desired, you may modify any of these workflows by going to Workflow Definition.

Workflow Generation - Create user type of workflows

Select the type of workflows to be created:

- Create user - Basic
Name:
- Create user - Advanced (with approval)
Name:
- Self registration for user (with approval)
Name:

Select target location:

Target:

5. If you select a Create workflow type, you can also specify one target location for the object the workflow creates.

The default target location is the searchbase for the COREid System.

6. Optionally, you can select additional attributes.

For a Create User, Create Group, or Create Object workflow, these attributes are entered during initiation or self registration steps.

For a Change Attribute workflow, these attributes are modified when running the workflow. A separate workflow is created for each attribute you select. For example, if you select five attributes, the QuickStart tool generates five change attribute workflows.

7. Click Generate.
8. View the summary report generated by the QuickStart tool.

Workflow Definition - QuickStart (2 of 2)

Summary report

The following workflows have been generated:

Workflow name	Warning (if any)
Create user - Advanced (with approval)	

Done

9. To test the workflow, click a link to one of the workflows on the summary report.

This initiates a workflow instance. For information on the process for using a workflow, see “How Users Access Workflows in a COREid Application” on page 175.

Note: To use a workflow as a portal insert, copy the resulting URL from your browser. See the *NetPoint 7.0 Developer Guide* for more information on creating portal inserts.

10. Click Done.

Creating a Self-Registration Workflow Using the Quickstart Tool

If you want to provide a user registration page for your Web portal, you can define a self-registration workflow and capture the resulting URL of the workflow. This URL can be used as a portal insert.

To define a self-registration workflow using the Quickstart tool

1. Create a self-registration workflow as described in “To define a workflow using the QuickStart tool” on page 191.
2. After clicking the Generate button, click the link for the newly-created workflow.
3. When the new workflow appears, copy the URL.

You can use this URL when you set up the user registration page for your Web portal. This URL is the link to the first page of the workflow. See “Creating a Self-Registration Workflow” on page 247 for other methods of defining self-registration workflows.

Using the Workflow Applet

In addition to using the QuickStart tool, you can define workflows using configuration pages that allow you to specify multiple options and subflows.

You must have permissions to define workflows. See “Delegating Administration” on page 47 for more information.

Typically, workflows contain at least two steps: one step to initiate the workflow and another step to commit the changes.

Task overview: Defining a workflow using the workflow applet

1. Invoke the Workflow Definition applet.

See “To access the Workflow Definition applet” on page 195 for details.

2. Select New to start creating a new workflow definition.

“Starting a New Workflow Definition” on page 196

3. If you selected a Create workflow type, identify a workflow target.

The target is the location in the directory tree where the object will be created. See “Defining an LDAP Target for Create Object Workflows” on page 199 for details.

4. Define a workflow step and action.

For each step in a workflow, there is an action. Actions are performed by people or by an automated method. You assign one action to each step in a workflow. You also assign participants to each step. See “Defining the First Step in a Workflow” on page 201 for details.

5. Associate attributes with the step.

Step actions are performed on one or more attribute values. These attributes may be taken from the directory or from an object template. See “Defining Step Attributes” on page 203 for details.

6. Define entry conditions for subsequent steps.

See “Defining Subsequent Steps” on page 206 for details.

7. Define the subsequent steps.

8. Define one or more subflows.

Subflows are conditions that must be satisfied for a particular step or workflow to complete. Like main workflow steps, subflows have associated actions, participants, and attributes. See “Defining a Subflow” on page 210 for details.

9. Define one or more commit steps to end the workflow.

If you are configuring a workflow using more than one schema (for example, LDAP and MIIS), you should configure separate commit steps for each schema type.

Note: In the 7.0 version of the COREid System, template attribute values can be sent to the back-end system used for provisioning, but these values cannot be read back in to COREid for display on profile pages. As a result, to check if a workflow configuration was done correctly and an instance of using the workflow was successful, you may have to examine the data in the back-end system.

To access the Workflow Definition applet

1. From the COREid System Console, select the User, Group, or Organization Manager.

If the Organization Manager has more than one tab, select the appropriate tab.

2. Click Configuration > Workflow Definition.

For the User Manager and Organization Manager, the following page is displayed:

Workflow Definition

Workflow Definition allows an administrator to define different workflow^s for different organizations. Different workflows pertain to different steps, attributes, and participants. To create a workflow using Quickstart [Click here](#)

The screenshot shows the 'Workflow Definition' applet interface. At the top, it displays '[1 of 3] Workflow Definition'. Below this, there is a header bar with a dropdown menu showing 'Change attribute - Advanced (with approval from Manager) - Out Of Office' and several action buttons: 'New', 'Modify', 'Copy', 'Delete', 'View', 'Enable', and 'Export All'. The main area is divided into four sections: 1) 'Workflow Name' with an empty text input field; 2) 'Workflow Type' with three dropdown menus: 'Create User', 'Change Attribute', and 'Audio', followed by a 'Use as Subflow' checkbox; 3) 'Description' with a large yellow text area and a vertical scrollbar; 4) 'Workflow Domain' with a list box containing 'dc=cc,dc=oblix,dc=net' and a vertical scrollbar. Below the list box is a 'Filter' input field containing 'dc=cc,dc=oblix,dc=net'. At the bottom of the applet, there are three buttons: 'Save Workflow', 'Cancel Workflow', and 'Next'.

3. If you are using the Group Manager, indicate the appropriate Group Type, if applicable. The following page shows a Group Type of Advanced Group. The available group types depend on your configuration, as described in “Adding Special-Purpose Object Classes to a Group Tab” on page 109.



Workflow Definition

[Next](#) [Cancel](#)

To create a workflow using Quickstart [Click here](#)

To create a workflow without optional group types, or if you have no group types defined, click Next.

To add predefined group types to your workflow, check them and then click Next.

Advanced Group

[Next](#) [Cancel](#)

4. If you are using the Group Manager, from the Workflow Definition page, select an appropriate group type if applicable and click Next.

If you do not select a group type, the Basic group type is used for this workflow.

Starting a New Workflow Definition

You can create workflow definitions for different sets of users. For example, you can define different Create User workflows for Engineering and Sales.

Note: For a simple Create User workflow, required attributes are Last Name and Name on most directory servers and Login ID if you use Active Directory.

To begin a new workflow definition

1. Invoke the workflow definition tool as described in “Using the Workflow Applet” on page 194.
2. Click New and wait for all buttons except the New button to become deactivated.
3. In the Workflow Name field, enter a name for your workflow.
4. From the Workflow Type drop-down, select the type of workflow you want to create.

For more information on workflow types, see “Workflow Types, Steps, and Actions” on page 179

If you are creating a subflow, see “Defining a Subflow” on page 210.

5. In the Workflow Description field, you can enter an optional description of this workflow.
6. In the Workflow Domain field, select the starting point in the directory tree from which this workflow is available.

Note: Do *not* use full LDAP URL while specifying the filter for workflow domain (or target domain) while creating the workflow. Only the LDAP filter is expected.

For instance, if you have different branches in your directory tree for Engineering and Sales, and you want this workflow to only apply to Engineering, you would select the top node for the Engineering branch of the directory tree. If you have a particularly flat directory tree or if the tree has a particularly high number of branches, you can narrow the workflow domain by entering an LDAP filter. See “Usage of Rules and Filters” on page 89. For example, if the starting point in the directory tree is `ou=people`, and you want to create a workflow just for administrators, you may want a filter that contains `(title=admin)`.

Note: Be sure to test performance when using filters. Filters are evaluated at run time, which can affect performance.

7. If you are in User or Organization Manager, click Next.

Depending on your workflow type, you are prompted to select a target as described in “Defining an LDAP Target for Create Object Workflows” on page 199, or you are prompted to define the first step in the workflow as described in “Defining the First Step in a Workflow” on page 201.

8. If you are in the Group Manager and you are working with an Oblix Advanced Group, specify the Subscription Type, if applicable.

For example, this might be the case when you define a step for selecting a group or for allowing a user to add themselves to a group. Subscription Type options are available to your participants if the `obGroupSubscriptionType` attribute was configured for the `oblixAdvanced Group` object class.

Workflow Definition

Workflow Definition allows an administrator to define different workflows for different organizations. Different workflows pertain to different steps, attributes, and participants.

[1 of 3] Workflow Definition

The screenshot shows the 'Workflow Definition' configuration window. At the top, there is a 'Workflows' dropdown menu set to 'No Workflows Specified' and a row of buttons: 'New', 'Modify', 'Copy', 'Delete', 'View', 'Enable', and 'Export All'. The main configuration area is divided into five sections:

- 1) Workflow Name:** A text field containing 'Create Advanced Group'.
- 2) Workflow Type:** A dropdown menu set to 'Create Group', followed by a 'Change Attribute' dropdown set to 'Business Category' and a checkbox for 'Use as Subflow'.
- 3) Description:** A text area containing 'Creates a group based on oblixAdvancedGroup Object Class'.
- 4) Workflow Domain:** A list box containing 'ou=Dealers', 'ou=Groups', 'ou=People' (highlighted), and 'uid=anonymous'. Below it is a text field containing the LDAP filter 'ou=People,dc=www,dc=oblix,dc=com'.
- 5) Subscription Type:** A section with three checkboxes: 'Closed', 'Open', and 'Open with Filter' (checked). Below these is a checkbox for 'Controlled through Workflow'.

At the bottom of the window are three buttons: 'Save Workflow', 'Cancel Workflow', and 'Next'.

The following subscription types are available:

Table 25 Workflow Subscription Types

Option	Description
No type selected	No subscription type is defined. Functionally equivalent to the Open policy.
Open	Enrollment is open to anyone who subscribes.
Open with Filter	Enrollment is open to any user who satisfies the Dynamic Filter (LDAP rule) for the group.
Controlled through Workflow	To subscribe or unsubscribe, the user must be the target of a select group step of a workflow.
Closed	Member list is closed. No changes are allowed. The default setting for the default_subscription policy parameter is SubscriptionPolicyClosed. This is located in <i>COREIDInstall_dir</i> /identity/oblix/data/common/groupdbparams.xml where <i>COREIDInstall_dir</i> is the directory where you installed the COREid system.

9. Click Next.

Depending on your workflow type, you are either prompted to select a target as described in “Defining an LDAP Target for Create Object Workflows” on page 199, or you are prompted to define the first step as described in “Defining the First Step in a Workflow” on page 201.

Defining an LDAP Target for Create Object Workflows

If you selected Create as the type of workflow you are defining, for example, Create User, you need to define one or more targets. The target is the location in the directory tree where the object will be created. For example, a target of `ou=bestmotors,o=company,c=us` allows objects to be created under the `ou=bestmotors` container. When a user is created using a workflow with this target, the directory entry may look like `cn=John Smith,ou=bestmotors,o=company,c=us`. If you define more than one target, the participant is presented with a drop-down selection list when the workflow is run. Workflow targets are always based on the LDAP directory tree. Targets cannot be based on a template schema.

If you are defining another type of workflow, clicking Next on the initial workflow definition page brings you to the step definition page described in “Defining the First Step in a Workflow” on page 201.

Note: The default only displays immediate child nodes of the searchbase. See “Modifying the Default Searchbase Scope” on page 162 for details.

To define a workflow target

1. If you have not already done so, start a new workflow as described in “To begin a new workflow definition” on page 196.
2. From the first Workflow Definition page, click Next.

The targets page appears, as illustrated below.

Workflow Definition

Workflow Definition allows an administrator to define different workflows for different organizations. Different workflows pertain to different steps, attributes, and participants.

[2 of 3] Target(s) Definition

Workflow Name : Create Advanced Group **Workflow Type :** Create Group

Target(s) GroupsBranch : ou=Groups,dc=wwm,dc=oblix,dc=com

New Delete

1) **Target Name**

2) **Target Domain**

- ⊕ cn=Test1
- ⊕ cn=weblogic_system
- ⊕ ou=Apps
- ⊕ ou=Dealers
- ⊕ **ou=Groups**
- ⊕ ou=People
- ⊕ uid=anonymous

ou=Groups,dc=wwm,dc=oblix,dc=com

Filter

Add

Save Workflow Cancel Workflow Previous Next

3. To define a new target, enter a name in the Target Name field.
For example, if you are creating a target for a dealership, the target name may be Dealer Name.
4. In the Target Domain field, select the location in the directory tree where the object will be created and click Add to add the target domain to the Target(s) field.

When you defined the workflow domain, you selected a branch of the directory tree that the workflow applies to. The target domain is a subset of the main workflow domain. You can use a filter to more closely specify the location for the target (any user object in the tree under the node you select).

Note: Do *not* use full LDAP URL while specifying the target domain (or workflow domain) while creating the workflow. Only the LDAP filter is expected. For example, `cn=Shutterbug Canavan` is expected rather than `ldap://ou=Partners,o=Company,c=US??sub?(cn=Shutterbug Canavan)`.

See “Usage of Rules and Filters” on page 89 for more information.

Note: If you added a filter for the workflow domain, you cannot specify a filter for the target.

5. Click Add.
6. To apply the workflow to additional targets:
 - Click New.
 - Supply another name and domain.
 - Click Add.
7. When you are done supplying target domains, click Next.

Defining the First Step in a Workflow

After naming the workflow and defining a target, if required, you are prompted to create the first workflow step. You will see a page similar to the following.

Workflow Definition

Workflow Definition allows an administrator to define different workflows for different organizations. Different workflows pertain to different steps, attributes, and participants.

Workflow Name : Create Advanced Group **Workflow Type :** Create Group

Defined Steps:

Step Properties

Select action to be performed*

To define the first step in a workflow

1. If you have not already done so, start a new workflow as described in “To begin a new workflow definition” on page 196.
2. If you have not already done so, for a Create workflow type, define a target as described in “To define a workflow target” on page 200.
3. In the Select action to be performed field, select an action.

For a Create Object workflow for the User or Organization Manager, the Initiate and Self Registration actions are available.

For a Create Object workflow for the Group Manager, the Initiate action is available.

4. Click Participants.

Most steps require participants to perform an action. The exception to this are steps with actions that occur automatically such as Commit and Enable, as well as External Action and the Self Registration action.

5. Use any of the following methods to specify participants.
 - **Roles**—Note that the role of Anyone refers to any user who is logged in to NetPoint. Roles are defined in the workflow parameter files `gsc_wf_param.xml`, `usc_wf_param.xml`, and `osc_wf_param.xml`. See “Customization of Data and Actions in a Workflow” on page 245 for details.

Note: If you chose Select Participants to Prenotify in the Select Participants field, do not choose Next Step Participants as the role. Also note that in the commit step for a Group Manager workflow, you should not check owner or member for post notification. There will be no email notifications for owner or member even if they are selected.

Participant roles (roles for people who can process a step) will only work after a commit, enable, or activate step has been completed. The commit, enable, or activate step creates the object's DN from which notification information can be determined.

- **Select Person**—See “The Selector” on page 40 for details on using the Selector. See “Search Filters for the Object Selector Display Type” on page 87 for information on how the selector may be configured.
- **Select Group**—See “The Selector” on page 40 and for details on using the Selector. See “Search Filters for the Object Selector Display Type” on page 87 for information on how the selector may be configured.
- **Build Filter**—See “Writing LDAP Filters Using Query Builder” on page 132 for information on creating an LDAP filter.

Workflow Name : Create User with LDAP filter Workflow Type : Create User

Defined Steps: [step 1] Initiate

New Modify Delete Step Insert Step

Step Properties

Action Subflows Attributes Participants Mail Notification

Select Participants*

Select Role: Annone

Build Filter: Build Filter

ldap://OU=Company,DC=qalab-fishcake,DC=obliz,DC=com??sub?(genbadg)

Select Person(s): Select User

<input checked="" type="checkbox"/>	Access Syposz	<input checked="" type="checkbox"/>	Annet Von Zuben	<input checked="" type="checkbox"/>	Bibi Humphrey
<input checked="" type="checkbox"/>	Blythè Eustis	<input checked="" type="checkbox"/>	Bàrrié Ariás	<input checked="" type="checkbox"/>	Bélna Dubosé
<input type="checkbox"/>	...	<input type="checkbox"/>	...	<input type="checkbox"/>	...

Select Group(s): Select Group

Save Step

Save Workflow Cancel Workflow Previous

6. Click Save or select step attributes as described in the following paragraphs.

Defining Step Attributes

Step actions are performed on one or more attribute values. When you configure a step action, you indicate if certain attribute values are required, and other configuration options. For example, on a Provide Information action, you can specify the mail attribute to ensure that the step participant is prompted to supply an email address.

Defining step attributes consists of the following:

- Selecting the attributes that should be available in this step of the workflow.
- Configuring attribute properties.

For attributes based on an object template (.tpl file), when you configure the attribute in the COREid System Console, it may be helpful to indicate the type of schema that the attribute belongs to. For example, for a workflow that provisions to MIIS, you may want to preface the attribute label with “MIIS.” This will be helpful when users view this attribute. Since the flow of data is one-way for provisioned attributes, the attribute values will not be displayed on the COREid profile page once the user submits the value. If your users have a question about this, the attribute label will help you determine if this is the expected behavior. See “Configuring Attributes” on page 81 for details.

To select attributes available for a step

1. If you have not already done so, start a new workflow as described in “To begin a new workflow definition” on page 196.
2. If you have not already done so, for a Create workflow type, define a target as described in “To define a workflow target” on page 200.
3. Begin defining a workflow step as described in “To define the first step in a workflow” on page 202.
4. After selecting participants for the workflow step, click Attributes.
5. From the Available Attributes panel, select one or more attributes to associate with the workflow step.

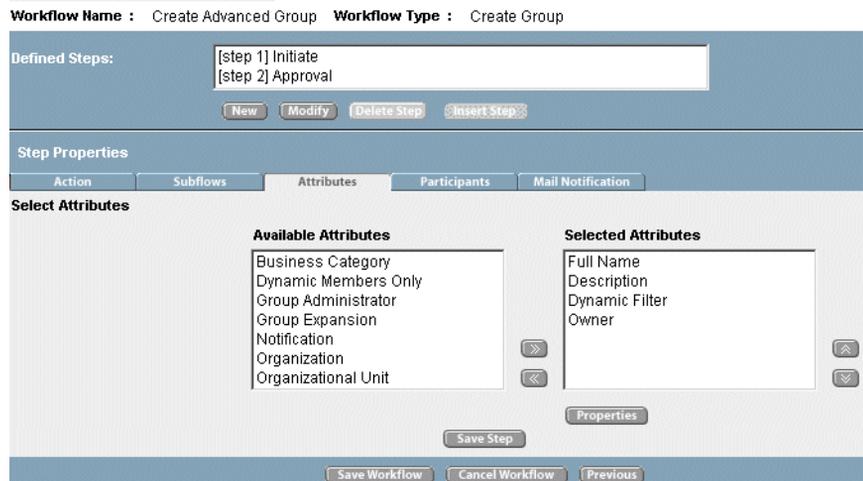
For a Change Attribute workflow, be sure that the topmost selected attribute has already been added to a panel on a profile page. This ensures that a “Request to Modify” button will appear on the appropriate profile page, enabling users to run instances of this workflow.

For information on making multiple selections, see “Keys for Selecting Multiple Attributes” on page 141.

6. Click >> to add the selected attributes to the Selected Attributes window.

By default for a Create Object workflow, the attribute that defines the Relative Distinguished Name (RDN) appears in the Selected Attributes window.

By default for a Change Attribute workflow, the attribute you selected as the basis for the workflow appears in the Selected Attributes window.



7. Save the step or configure attribute properties, if applicable, as described in the following paragraphs.

Note: You cannot save a workflow until all required attributes (as defined in the object class schema) are configured for the workflow.

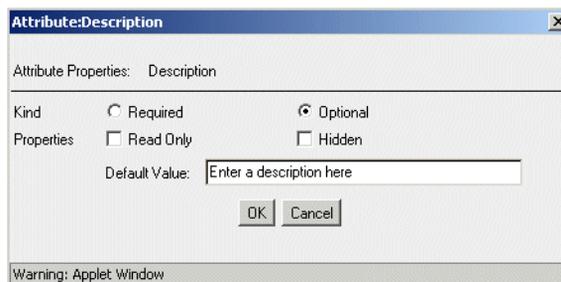
To configure attribute properties

1. From the Selected Attributes window, select one or more attributes that you want to configure.

For information on making multiple selections, see “Keys for Selecting Multiple Attributes” on page 141.

2. Click Properties.

An Attribute Properties dialog appears:



3. Select one or more properties for these attributes:

- **Required**—The workflow participant must provide a value for this attribute.

Note: A required attribute cannot be hidden or read-only.

- **Optional**—The workflow participant may provide a value for this attribute.
- **Read-only**—The workflow participant can see but cannot modify the attribute.
- **Hidden**—The workflow participant cannot view this attribute value. The attribute is available in the Identity Event Plug-In API and IdentityXML.
- **Default value**—Displays a text string. This text string should helpful information for the participant. For example, a string showing the correct format for entering a phone number could be the default value for the phoneNumber attribute. The value is limited to text display types.

4. Click OK.

5. Click Save.

You can now define Mail Notification participants, or you can define additional steps for this workflow.

Note: When you define Mail Notification participants, if you chose Select Participants to Prenotify in the Select Participants field, do not choose Next Step Participants as the role. Also note that in the Commit step for a Group Manager workflow, you should not check Owner or Member for post notification. There are no email notifications for owner or member even if they are selected.

A commit, enable, or activate step must be completed for a role selected for pre or post notification to work. Before the commit, enable, or activate step is completed, the object exists only in the workflow instance information in the Oblix tree. The commit, enable, or activate step creates the object's DN from which notification information can be determined.

Note: For information on customizing email notifications, see the *NetPoint 7.0 Customization Guide*.

Defining Subsequent Steps

A workflow consists of at least an initiating step and a completion step, and may have more steps and subflows. As part of the procedure for creating a second (or third, or more) step in a workflow, you define entry conditions for the step. Entry conditions consist of:

- Identifying what step precedes this one.

- Identifying the required outcome for the previous step.

To define subsequent steps in a workflow

1. After completing the first step in a workflow, as described in “To define the first step in a workflow” on page 202, click New.

New fields appear on this page appropriate for configuring subsequent steps in a workflow.

2. From the Previous Step drop-down list, select the step that should precede this action.
3. From the Return Value drop-down list, indicate whether the previous step should return a value of true or false in order for this action to execute.

You will want the previous step to return a value of True if it completes successfully. Select False to generate an error report when a previous step returns a value of false. Situations that return a value of false include:

- A participant rejects a workflow ticket
- The commit step fails
- The Identity Event Plug-in API or IdentityXML forces a return value of false.

Workflow Name : Create Advanced Group Workflow Type : Create Group

Defined Steps: [step 1] Initiate
 [New] [Modify] [Delete Step] [Insert Step]

Step Properties

Action Subflows Attributes Participants Mail Notification

Select entry conditions*

Previous Step* 1:Initiate

Return value* true Wait for Subflows [Add]

Selected Value
 1:true:true
 [Delete]

Select action to be performed* Approval

[Save Step]

[Save Workflow] [Cancel Workflow] [Previous]

4. In the Action drop-down list, select the action. You may choose enable, activate, and other actions.

The available actions depend on the action in the previous step. Examples:

- Provide Information cannot precede Initiate.
- An error report action usually provides a reason for a step failure. For example, rejection of an attribute value or denying a user activation request.

- Usually a condition of false is the entry condition for an error report step. For example, if a participant in the step prior to the error report step rejects an activation operation, the workflow may proceed to an error report step.

Note: For workflows used for provisioning, you should have at least two commit actions defined per workflow, one to commit (or enable, or activate, and so on) the data in LDAP, the other to write the provisioning data.

5. Select Wait for Subflows to delay execution of this action until all subflows from preceding steps are complete regardless of their return value.

Selecting this check box appends the return value entry condition with `:true`. If you do not select the checkbox, the return value entry condition is appended with `:false`. See “Defining a Subflow” on page 210 for more information.

6. As needed, add participants and configure attributes as described in “Defining the First Step in a Workflow” on page 201.
7. Save the workflow.

Committing Workflow Steps

The last step of a workflow commits the data to a particular schema domain. By default, the schema domain is the LDAP directory that the COREid System communicates with. However, if you have configured template attributes in a workflow, you must configure a separate step to commit the data to the schema domain for the template attributes.

Commit steps for attributes in template schema domains can be processed by the Identity Event API and passed to back-end systems for provisioning.

Oblix also supplies an agent for MIIS. This agent is installed as part of the COREid System. Attribute values that are committed to the MIIS schema domain are automatically processed by the MIIS Agent. See the *NetPoint Integration Guide* for details.

Enabling the Workflow

You must enable workflows before they are made available to participants in the NetPoint COREid System or to external applications.

To enable the workflow

1. Access the User, Group, or Organization Manager.
2. Click Configuration > Workflow Definition.
3. From the workflows drop-down menu, select the workflow.
4. Click Enable.

Testing the Workflow

You must enable the workflow before you can test it. See “Enabling the Workflow” on page 208 for more information. You must also be a workflow participant to be able to test it.

To test the workflow

1. From the COREid System Console, select the application in which an instance of this workflow can be run.

For example, for a Create User workflow, you would open the User Manager.

2. To initiate the workflow, select the function associated with the workflow.

For example, for a Create User workflow, you would open the User Manager and select the Create User Identity function. Test this workflow by creating a new user. If more than one workflow has been defined for creating a user, select this workflow type from the drop-down list that will appear on the Create User page.

For Change User workflow, you would select the Request to Modify function in the User Manager, and so on.

To run a workflow in the Group Manager

1. From the COREid System Console, select Group Manager.
2. Select the group type panel corresponding to the group type in the workflow definition.

For example, you may have different group type panels for the structural group object class and the oblixAdvancedGroup object class. See “Adding Auxiliary and Template Object Classes to a Tab” on page 108 for details.

Example of Defining a Workflow

The following is an example of defining a Create User workflow. In this example, you define a workflow that allows anyone who is logged in to NetPoint to create a user. This workflow generates a ticket requesting a name and email address to be provided for this user. When processed, the new user is enabled in NetPoint.

To create this workflow

1. From the User Manager, invoke the Define Workflow page and select Create User as the workflow.
2. Name this workflow Test New User Creation Workflow.
3. On the Target DN page, accept the defaults by clicking Add.
4. Click Next to proceed from the Target Domain page to the Workflow Definition Page.

5. Create an Initiate step for the workflow.
Click the Participants tab and define the participants to be the role of Anyone.
Click the Attributes tab and select the attributes of Name and Email as the information to be provided.
6. Click New to create a new step with the Enable action type.
7. Save and enable the workflow.
8. Test the workflow by trying to create a new user in the User Manager.

Defining a Subflow

Only the Change Attribute workflow type can be a subflow.

To create a subflow

1. Create a new workflow, as described in “Using the Workflow Applet” on page 194.
2. In the Workflow Name field, enter a name for your workflow.
3. From the Workflow Type drop-down list, select Change Attribute.
4. Specify the Change Attribute.

This is the attribute that the workflow modifies. You can select additional attributes on the next page.

Workflow Definition

Workflow Definition allows an administrator to define different workflows for different organizations. Different workflows pertain to different steps, attributes, and participants. To create a workflow using Quickstart [Click here](#)

[1 of 2] Workflow Definition

The screenshot shows the 'Workflow Definition' applet interface. At the top, there is a 'Workflows' section with a dropdown menu showing 'Auto generated NetPoint BEA Realm create user workflow [Disabled]'. Below this are buttons for 'New', 'Modify', 'Copy', 'Delete', 'View', 'Enable', and 'Export All...'. The main configuration area includes:

- 1) Workflow Name:** A text field containing 'Change Login & Password'.
- 2) Workflow Type:** A dropdown menu set to 'Change Attribute', followed by a text field 'Change Attribute', another dropdown menu set to 'Login', and a checked checkbox 'Use as Subflow'.
- 3) Description:** An empty text field.
- 4) Workflow Domain:** A list box containing 'dc=wwm,dc=oblrix,dc=com'. Below the list box is a 'Filter' text field containing the same domain string.

At the bottom of the applet are buttons for 'Save Workflow', 'Cancel Workflow', and 'Next'.

5. If you want this workflow to be initiated from another workflow, select Use as Subflow.

6. Complete the rest of the workflow as you would any other workflow.

Note: All subflow definitions must contain an approval step action.

Associating a Subflow with a Workflow

You must associate a subflow with a specific workflow step in the main workflow. During workflow runtime, any subflows configured for a specific step will be invoked *after* the step action has executed.

To associate a subflow with a workflow

1. Invoke the workflow application, as described in “To access the Workflow Definition applet” on page 195.
2. Select a workflow to which you want to assign a subflow.
3. Click Modify.

The page refreshes and shows the step definitions page.

4. Click the Subflows tab.
5. In the Defined Steps area of the page, select the place in the workflow sequence where you want to insert a subflow.
6. In the Select Subflows area of the Subflows tab, select the subflow that you want to be a part of this workflow.

The screenshot shows a web interface for configuring a workflow step. At the top, it displays 'Workflow Name : Create WWM User' and 'Workflow Type : Create User'. Below this, the 'Defined Steps' section shows a text input field containing '[step 1] Initiate' and four buttons: 'New', 'Modify', 'Delete Step', and 'Insert Step'. The 'Step Properties' section has five tabs: 'Action', 'Subflows', 'Attributes', 'Participants', and 'Mail Notification'. The 'Subflows' tab is active, showing a 'Select Subflows' area. This area is divided into 'Available Subflows' and 'Selected Subflows'. In the 'Available Subflows' list, 'LoginSubflow' is highlighted. To the right of this list are two arrows: a right-pointing arrow (labeled '>>') and a left-pointing arrow (labeled '<<'). The 'Selected Subflows' list is currently empty, with a right-pointing arrow (labeled '>') above it and a left-pointing arrow (labeled '<') below it.

Select the subflow(s) you want to assign to this step and click >>.

Note: If you do not see your subflow here, verify that it is marked as a subflow, and it is enabled. The attribute that is the target of the subflow cannot also be used in the workflow to which the subflow is assigned.

7. Save the step.
8. On the subsequent step(s), you may optionally indicate whether or not you want to wait for subflows to complete.
 - a) Select the step that should be delayed until the subflow is complete and click Modify.

- b) Click the Wait for Subflows checkbox.

Approving Subflow Steps

The Subflow Approval step reports on the current status of subflows triggered from the main flow. By default, the status is set to Approved or Rejected during either an Approval step or a Provide Approval step. This step also allows for the configuration of attributes.

Note: You can set the subflow status programmatically using Identity Event Plug-in API or IdentityXML. Oblix IDLink requires a Subflow Approval step to implement changes on the ESS server. If an Oblix IDLink subflow fails, all subflows are rejected. See the *NetPoint 7.0 Customization Guide* for information on the Identity Event Plug-in API. See also the *Oblix IDLink 1.0 Administration Guide* for information.

Advanced Workflow Ticket Routing

Ordinarily, the static participants you specify when you create a workflow step are the users responsible for completing that step. To avoid processing bottlenecks or to ensure that each ticket reaches the participant most appropriate to process it, the following three advanced ticket routing features enable the replacement of static participants under specific circumstances:

- Instead of specifying static participants when you create a workflow, you can have a workflow plug-in or application choose dynamic participants according to runtime conditions.
- If a static or dynamic participant is going to be out of the office or otherwise unable to process workflow tickets, he or she can set an Out of Office flag in his or her user profile so that all incoming tickets are redirected to a surrogate participant for as long as the flag remains activated.
- If the participants receiving a given workflow ticket fail to process it within a specified interval, that ticket can be sent to an escalation participant, who assumes full responsibility for the ticket.

Configuring Workflow Actions for Advanced Ticket Routing

Not all workflow steps can be configured for advanced ticket routing. For instance, the first step in a workflow can never be rerouted. Fortunately, it is never necessary to reroute the first step, because the person who initiates the workflow is also the participant for the first step, which is always workflow initiation.

Steps that do not involve user action cannot be rerouted since, by definition, they do not involve user participants. For instance, a step involving the automatic retrieval of provisioning data from an external database involves no human participants, and therefore, participant replacement is moot.

The following table lists the user actions that can be associated with workflow steps:

Table 26 User Actions Available for Advanced Ticket Routing

User Action	Availability
Approval	Available by default for dynamic participants, surrogates, and time-based escalation
Provide information with approval	
Initiate	Available by default for dynamic participants and surrogates; can be enabled for time-based escalation by modifying the appropriate workflow parameter file. For details, see "To modify the workflow parameter files" on page 226.
Self registration	
Provide information	
Subflow approval	
Activate	
Deactivate	
Error report	
Select groups	
Requests	
Change information	
Change information with approval	

About Notifying Newly Assigned Step Participants

The Mail Notification tab in the workflow applet enables you to configure email notification for participants who are assigned tasks to complete when workflow tickets are rerouted. You can configure mail notification for each step to which ticket rerouting can apply

To configure mail notification for any step involving advanced ticket rerouting, complete the following procedure:

To configure email notification for steps involving advanced workflow ticket rerouting

1. As appropriate for the workflow you are modifying, navigate to the User, Group, or Organization Manager > Configuration > Workflow Definitions.
2. Select the Workflow you wish to modify, then click Modify.
3. If you are modifying a Change Attribute workflow, click Next once. For any other type of workflow, click Next twice.
4. Select the step for which you wish to set mail notification, then click Modify.
5. Click the Mail Notification tab.
6. To enable notification for static, dynamic, and surrogate participants, click Select Participants to Prenotify.
7. Specify by Person, Group, Role, or Rule, the users to notify as described in “Use any of the following methods to specify participants.” on page 202.
8. If you also need to notify escalation participants, click Select Escalation Notifies, then repeat step 7.
9. Click Save Step to commit your step-specific changes.
10. Click Save Workflow to save the entire workflow.

Specifying Dynamic Participants

The dynamic participants feature is one of the advanced workflow options that enable the automatic routing of workflow tickets to alternate participants, as determined by circumstances at runtime.

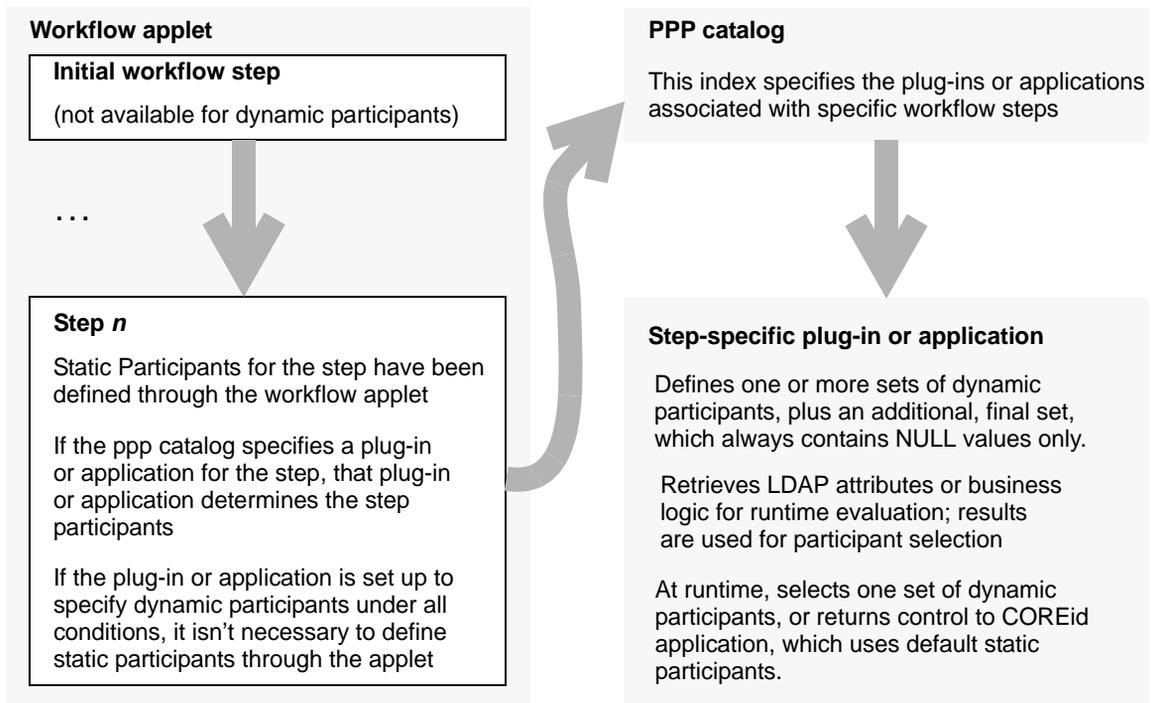
About Workflow Participants

NetPoint supports the following two types of workflow participants:

- *Static participants* are specified through the workflow applet, usually when you define a workflow step. For example, when you create a workflow to set up network accounts for new employees, you can include an approval step that routes all incoming tickets to the network security manager. Unless you later add a workflow plug-in or application, this pre-designated static participant serves as the primary recipient for all tickets generated by this workflow step.
- *Dynamic participants* must be specified in a workflow plug-in or application, rather than through the workflow applet. These conditional participants are selected on the basis of runtime attribute values or external business logic. For example, your plug-in can specify that all purchase requests greater than \$2,500 go to the accounting manager, all requests smaller than \$50 go to the petty cash clerk, and all other requests go to any available staff accountant.

About Workflow Ticket Routing

As the following graphic illustrates, when a workflow is run and program execution reaches a step that has been enabled for dynamic participants, a workflow plug-in or application selects a set of dynamic participants and sends them a workflow ticket. In cases where the plug-in or application does not select any dynamic participants, the calling application sends the ticket to the original static participants, just as if the workflow plug-in or application had never intervened.



About Dynamic Participants

You designate dynamic participants using the same criteria you use to specify static participants. This includes specification by person, by group, by role, and by rule. For details, see the procedure step “Use any of the following methods to specify participants.” on page 202.

You can define dynamic participants for every step in a workflow, except for the first step, which must be initiated by a static participant.

When step instances are assigned at runtime, dynamic participants inherit the same ticket-processing rights that static participants would normally receive. These rights extend only to tickets specifically assigned to the dynamic participant. In other words, the dynamic participant does not receive all the rights assigned to the original participant, as would be the case if the Substitute Rights feature were used to create a delegate. See “Adding Substitute Administrators” on page 55.

Since the identity of the dynamic participants is not known until the associated workflow step is run, they are not available for certain workflow services, such as post-action email generated by the previous workflow step. However, it is possible to notify dynamic participants through pre-action email generated by the workflow step that selects them. See the procedure “To prepare a workflow step for dynamic participants” on page 218.

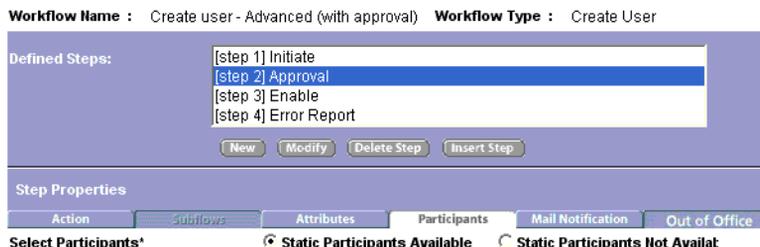
About Static Participants

Under normal circumstances, workflow steps use static participants, whom you designate through the workflow applet.

In cases where the ppp catalog specifies a plug-in or application for a given step, the plug-in or application receives the first opportunity to select dynamic participants according to the run-time values it evaluates. Only if the plug-in or application declines to specify a set of dynamic participants does control pass back to the main application, where the static participants are specified as the primary step participants.

About the Static Participants Not Available Button

If you know in advance that a workflow plug-in or application will always select dynamic participants for a given step, you don’t have to define static participants for that step. Remain aware, however, that if you do elect not to specify static participants, you must toggle the default Static Participants Available button to Static Participants Not Available. As the following graphic illustrates, these radio buttons appear on the Participants tab of the workflow applet, which is accessible through the User, Group or Organization Manager, as appropriate to the workflow you are configuring.



Enabling Dynamic Participants

The dynamic participants feature is not enabled by default. To activate this feature, you must complete the procedures listed in the following task overview:

Task overview: Assigning dynamic participants to a workflow step

1. Use either the QuickStart tool or the workflow applet to create a workflow containing the steps that will utilize dynamic participants. See “Using the QuickStart Tool” on page 190 and “Using the Workflow Applet” on page 194.
2. If any possibility exists that a given step will use static participants, you must define static participants for that step. See the procedure step “Use any of the following methods to specify participants.” on page 202.

On the other hand, if you know in advance that a given step will always use dynamic participants, you don’t need to define static participants for that step. See “About the Static Participants Not Available Button” on page 216.

3. If you wish, set up email notification for dynamic participants. For details, see step 5 of the procedure “To prepare a workflow step for dynamic participants” on page 218.
4. Add a pointer to the ppp catalog file so that the proper plug-in or application is called at runtime to select dynamic participants. For details, see “To modify oblixpppcatalog.lst” on page 218.
5. Create a pre-action plugin or application to select dynamic participants at runtime. A typical plug-in or application might contain code sections to perform the following:
 - Specify at least three sets of dynamic participants, the last of which always contains NULL values only
 - Specify attributes or business logic to be evaluated at run time
 - Select dynamic participants on the basis of the evaluation
 - Ensure that the selected participants actually exist in the NetPoint directory; otherwise, the dynamic participant selection process will fail, but the workflow engine will not return an error message
 - Pass the list of dynamic participants back to the calling COREid application

For details, see “Task overview: Creating a plugin or application to select dynamic participants” on page 220.

Important: Plug-ins and applications that enable dynamic participants must be of type “pre-action.” They can never be of type “post-action.”

To prepare a workflow step for dynamic participants

1. In the User Manager, Group, Manager, or Organization Manager, navigate to Configuration > Workflow Definitions.
2. From the list marked Workflows, select the workflow containing the step being prepared for dynamic participants, then click Modify.
3. If any possibility exists that the current workflow step can ultimately use static participants, define static participants by Role, Rule, Persons, or Group, as described in “Use any of the following methods to specify participants.” on page 202.

If the preceding condition does not apply, or you previously defined static participants for this step, proceed directly to step 4.

4. If the current workflow step will use a plug-in or application to specify dynamic participants, and there are no conditions under which static participants can ultimately receive the workflow ticket for the current step, activate the Static Participants Not Available button, as described in “About the Static Participants Not Available Button” on page 216. Otherwise, proceed directly to step 5.
5. If you wish, set up email notification by clicking the Mail Notification tab, clicking the Select Participants to Prenotify button, then selecting Current Step Participants in the Select Role box. If they are ultimately selected, the dynamic participants will receive e-mail announcing that workflow tickets have been assigned to them.

Note: The Select Participants to Prenotify button turns on email notification for static participants when the Static Participants Available switch is active. By contrast, it sends notifications to dynamic participants when the Static Participants Not Available switch is active.

6. Commit the step-specific information by clicking Save Step and then clicking OK to dismiss the pop-up that asks you to confirm the operation.
7. Commit the information pertaining to the entire Workflow by clicking Save Workflow and then clicking OK to dismiss the pop-up seeking to confirm the operation. If an additional pop up asks whether you want to enable the workflow, click Yes.

To modify oblixpppcatalog.lst

1. Complete the following sub-steps to determine the workflow ID of the workflow containing the step for which you wish to set dynamic participants:
 - a) Launch the User, Group, or Organization Manager, as appropriate for the workflow you wish to modify.
 - b) Navigate to Configuration > Workflow Definitions.

- c) Select the workflow you wish to modify, then click View.
- d) Make a note of the value reported in the Workflow Definition View pop-up for `obworkflowid`, which will appear in a string similar to the following:

```
Workflow DN : obworkflowid=5985de47196a4a728a629a429b6a5194
```

2. Use any plain-text editor to launch the file `oblixpppcatalog.lst`, which is located in the following directory:

```
Component_install_dir\identity\oblix\apps\common\bin
```

where `Component_install_dir` is the root installation folder for the COREid Server that runs your workflow.

3. Add one of the following strings to `oblixpppcatalog.lst`:

```
obworkflowid_workflowstep_preaction;lib;
Component_install_dir\identity\oblix\path
\pluginName.dll;functionName;
```

or

```
obworkflowid_workflowstep_preaction;exec;
Component_install_dir\identity\oblix\path
\applicationName.exe;functionName;
```

where:

- `obworkflowid` is the workflow identification number you noted in step 1d of this procedure
- `workflowstep` is the step for which you wish to define dynamic participants
- `path` is the path under `Component_install_directory\identity\oblix\` leading to `pluginName.dll` or `applicationName.exe`, which are the code objects that select the dynamic participants at runtime

If you specify a plug-in, you must also specify `functionName`, which is the function within the dynamic link library plug-in that sets the criteria for the dynamic participants.

You insert the first line of code if you are using a plug-in to specify the dynamic participants; you insert the second line if you are using an executable program, instead of a plug-in.

In any case, the line you insert must end with a semi-colon, and you may place it anywhere in the `oblixpppcatalog.lst` file, as long as that line does not interrupt an existing line.

The line you insert should be similar to the following:

```
wfqs20040901T17251953156_2_preaction;lib;
Component_install_dir\identity\oblix
\unsupported\ppp\ppp_dll\ppp_dll.dll;
WorkflowPreActionSetDynamicParticipantsTest;
```

4. Save oblixpppcatalog.lst in its original location.

Task overview: Creating a plugin or application to select dynamic participants

1. Use C++ to create a plugin or application that specifies the dynamic participants selected when program execution reaches the workflow step you specified in step 3 of “To modify oblixpppcatalog.lst” on page 218.
2. You can use various conjunctions of LDAP attributes or proprietary business logic to specify the conditions under which one group of dynamic participants is chosen over the others at runtime.
3. Include in the plugin or application, the following header files, which enable the pre-action processing necessary for dynamic participant selection:

```
obppp.h  
obpppwf.h  
obpppdata.h
```

4. Within the application or plug-in, define three or more sets of dynamic participants using any combination of roles, rules, persons, or groups. The final item in each array must always be NULL. For instance:

- a) If you wish to specify persons, insert lines similar to the following:

```
PPPSetVals[0] = "cn=Van Oman, ou=Sales, ou=Dealer1k1,  
ou=Latin America, ou=Ford, o=Company,c=US";  
PPPSetVals[1] = "cn=Fabien Esser, ou=Sales, ou=Dealer1k1,  
ou=Latin America, ou=Ford, o=Company,c=US";  
PPPSetVals[2] = NULL;  
data->Set("DynamicParticipant.Persons", PPPSetVals);
```

- b) If you wish to specify groups, insert lines similar to the following:

```
PPPSetVals[0] = "cn=Basic group1k1, ou=Groups, ou=Dealer1k1,  
ou=Latin America, ou=Ford, o=Company,c=US";  
PPPSetVals[1] = "cn=Basic group1k2, ou=Groups, ou=Dealer1k1,  
ou=Latin America, ou=Ford, o=Company,c=US";  
PPPSetVals[2] = NULL;  
data->Set("DynamicParticipant.Groups", PPPSetVals);
```

- c) If you wish to specify roles, insert lines similar to the following:

```
PPPSetsVals[0] = "ob_self";  
PPPSetsVals[1] = "manager";  
PPPSetsVals[2] = NULL;  
data->Set("DynamicParticipant.Roles", PPPSetsVals);
```

Remember, of course, that only certain roles are valid for particular workflow types. See page 202.

- d) If you wish to specify rules, insert lines similar to the following:

```
PPPSetsVals[0] = "(cn=rohit*)";  
PPPSetsVals[1] = "(cn=beth*)";  
PPPSetsVals[2] = NULL;  
data->Set("DynamicParticipant.Rules", PPPSetsVals);
```

Specifying Surrogates

You can configure the COREid applications so that when a static or dynamic participant is not available to perform the actions assigned for a particular workflow step, that participant can set an Out of Office flag in his or her user profile, causing incoming workflow tickets to go to one or more designated surrogate participants. The surrogate is granted whatever rights the original participant had to process the rerouted tickets.

Only tickets created after activation of the Out of Office flag are rerouted to the surrogate. The original participant retains responsibility for processing all tickets created prior to activation of the Out of Office flag.

When the Out of Office flag is turned on, it applies to all of the steps in all of the workflows for which the participant has been designated as a static participant or potential dynamic participant.

When the Out of Office flag is reset to Off, newly created tickets are once again routed to the original participant. The surrogate retains responsibility for processing all tickets routed to him or her while the Out of Office flag was active, but no new tickets are routed to the surrogate unless the original participant once again activates his or her Out of Office flag.

The same workflow applet setting that sends ticket assignments to original participants also notifies surrogates and others that the workflow ticket has been rerouted because of the Out of Office flag.

Task overview: Enabling surrogates

1. Associate the attribute of your choice with the Out of Office semantic type through the Common Configuration tab of the COREid Console. You only

need to do this once. See “To associate an Out of Office attribute with the Out of Office semantic type” on page 222.

2. Specify one or more surrogates through the Out of Office tab in the workflow applet. See “Use any of the following methods to specify participants.” on page 202.
3. Individual users activate their Out of Office flags in their User Profiles. See “To activate the Out of Office flag” on page 223.

To associate an Out of Office attribute with the Out of Office semantic type

1. Choose an attribute in the LDAP directory to associate with the Out of Office semantic type. It must be an attribute with a boolean value that indicates whether the user is in or out of the office.

For convenience, Oblix supplies the attribute `obOutOfOfficeIndicator`, but you can use any suitable attribute in your directory.

2. Navigate to COREid Console > Common Configuration > Configure Object Class > `gensiteorgperson` > Modify Attributes.

Logged in user: Master Admin

System Admin | User Manager Configuration | Group Manager Configuration | Org. Manager Configuration | Common Configuration

▼ Configure Object Class

Configure Workflow Panels

Configure Master Audit Policy

Configure Global Auditing Policies

Modify attributes

Through Modify attributes, you can modify or configure the display name, semantic type, display type, and attribute value(s) for the attributes in the `genSiteOrgPerson` object class. After modification, please click done button to save the attribute information.

Attribute	<input type="text" value="0"/> obdirectreports obindirectmanager oblocationdn oboutofofficeindicator obrectangle	Display Name	<input type="text" value="Out Of Office Indicator"/>
Data Type	String(Case-insensitive)	Semantic Type	(None)
Display Type	Boolean	Attribute Value(s)	<input checked="" type="radio"/> Single <input type="radio"/> Multiple

No Display Type Properties Available

Save Done Reset

3. From the attribute list, select the attribute you wish to associate.
4. In the Semantic Type box, select “Out of Office - Indicator.”
5. Make sure that Boolean is selected in the Display Type box, then click Done to commit the change.

Note: This procedure only needs to be performed once.

To specify a surrogate

1. As appropriate to the particular workflow containing the step for which you wish to specify one or more surrogates, log onto the User, Group, or Organization Manager, then navigate to Configuration > Workflow Definitions.
2. Select the workflow containing the step for which you wish to specify surrogates and click Modify.
3. Click Next once if you are modifying a Change Attribute workflow. Click Next twice if you are modifying any other type of workflow.
4. Select the step for which you wish to specify surrogates, then click Modify.

You can specify a surrogate for any workflow step associated with a user action.

5. Click the Out of Office tab.
6. Specify one or more Surrogates using any combination of the Person, Group, Role, and Rule tools. See “Use any of the following methods to specify participants.” on page 202.

The Select Indirect Roles box provides check boxes to select whatever roles are currently defined in your directory. These roles are considered indirect, because they apply to the current participant, rather than the workflow target.

7. Click Save Step to commit the changes for that step.
8. Repeat the preceding steps for each workflow step for which you wish to specify a surrogate.
9. Click Save Workflow to save the entire workflow.

To activate the Out of Office flag

1. Verify that you, as a potential static or dynamic user, have been granted sufficient privileges (search, read, and write) to perform the operations described in this procedure.
1. Navigate to User Manager > My Identity, then click Modify.
2. In the Personal Information section, toggle the Out of Office Indicator to True. (This attribute is False, by default).
3. Click Save to commit the change, then click OK to dismiss the pop up that seeks to confirm the operation.

Enabling Time-based Escalation

If the participant or participants assigned to process a workflow ticket do not do so within a specified interval, you can have NetPoint “escalate” the ticket by rerouting it to a different participant. The original participant can no longer process the escalated ticket: it must now be processed by the escalation participant, who receives all rights previously given to the original participant for processing the ticket.

If the escalation participant does not process the ticket within the allotted time, the ticket is escalated again, and so on, until it is escalated to the NetPoint administrator, who is the last possible participant to be assigned responsibility for the escalated ticket.

By default, you can enable time-based escalation on any workflow step, provided the following two conditions hold true:

- The escalated step is not the initial step in the workflow
- The action associated with the step has been enabled for escalation. By default, only Approval and Provide Information and Approval are enabled for escalation, but you can enable other actions by adding lines to the appropriate workflow parameter file. See the procedure “To modify the workflow parameter files” on page 226 for details.

To enable time-based escalation

1. As appropriate to the particular workflow for which you wish to set up time-based escalation, log onto the User, Group, or Organization Manager, then navigate to Configuration > Workflow Definitions.
2. Select the workflow for which you wish to set up escalation and click Modify. If a pop up appears to warn you that only certain settings can be modified while pending tickets exist for the workflow, dismiss it by clicking OK. If you are modifying a Change Attribute workflow, click Next once; if you are modifying any other type of workflow, click Next twice.
3. Highlight the step for which you want to enable escalation, then click Modify.
The step you select must be associated with an action that is enabled for escalation. By default Approval and Provide Information and Approval are enabled. To enable additional actions to support time-based escalation, see the procedure “To modify the workflow parameter files” on page 226 for details.
4. Click the Escalation tab.

- Specify the interval after which the ticket will be escalated. You can set the interval in days, minutes, or hours. This interval applies to all escalation levels.

Workflow Name : Change attribute - Advanced workflow (with approval from Group Owner) - Notification - Groups **Workflow**

Defined Steps:

- [step 1] Request
- [step 2] Approval
- [step 3] Commit
- [step 4] Error Report

New Modify Delete Step Insert Step

Step Properties

Action Subflows Attributes Participants Out of Office Escalation Mail Notification

Escalate ticket after 1 Day(s) of idleness for 3 levels

Route to participants

Select Indirect Role

- Primary Surrogate Indirect Manager Manager Admin
- Person can view personal info

Build Filter

Select Person(s)

Select Group(s)

Save Step

Save Workflow Cancel Workflow Previous

- Specify the participant or participants to whom the ticket will be escalated. Roles are “indirect” in the sense that they are evaluated not with respect to the workflow target, but with respect to the participant who did not process the ticket in time to prevent the most recent escalation. For instance, if Manager is checked in the Select Indirect Roles box, and the accountant who initially receives the ticket does not process it quickly enough, the ticket is escalated to that accountant’s manager (and not the manager of the workflow target).
- Specify the number of times (levels) a ticket can be escalated. This does not include the final escalation level, which always routes the ticket to the NetPoint administrator.

You can only specify one set of escalation participants. This single set applies to all escalation levels. If you specify a unique user, for example, the ticket is escalated to that person each time escalation is triggered. If that escalation participant does not process the ticket at any level, the ticket is ultimately escalated to the NetPoint Manager.

If, on the other hand, you specify a role that is held by a different person at each level, the ticket will be escalated to a different person at each level. For instance, if you specify Manager, the ticket will be escalated to the manager of the person to whom the ticket was originally issued, then to the manager of that manager, and then to that manager’s manager, and so on.

- Click Save Step, then click Save Workflow to commit the setting you have entered on the escalation tab.

- Specify the people who will be notified about the escalation by clicking the Mail Notification tab.

Workflow Name : Change attribute - Advanced workflow (with approval from Group Owner) - Notification - Groups **Workflow**

Defined Steps:

- [step 1] Request
- [step 2] Approval
- [step 3] Commit
- [step 4] Error Report

New Modify Delete Step Insert Step

Step Properties

Action Subflows Attributes Participants Out of Office Escalation Mail Notification

Select Participants* Select Participants to PreNotify Select Participants to PostNotify

Select Escalation Notifies

Select Role

Owner Member Group Administrator

Build Filter

Select Person(s)

Select Group(s)

Save Step

Save Workflow Cancel Workflow Previous

- Click Select escalation notifiees.
- Select the people to be notified by Person, Group, Role, or Rule. The available roles are the following:
 - Previous step owners**—This is the participant who completed the previous step.
 - Current step participants**—These are the people currently assigned to process the just-escalated ticket.
 - Next step participants**—These are the people who will be assigned to process the next step. Only the static participants defined for the next step can be notified, since the email notifications are sent before the execution flow reaches the next step, and the identity of the dynamic participants is determined.
 - Initiator**—This is the user who initiated the work flow.

To modify the workflow parameter files

- Complete this procedure only if you want to enable time-based escalation for a user action other than Approval or Provide Information and Approval. For a

list of the user actions that can be enabled for time-based escalation, see Table 26 on page 213.

- Using any plain text editor, launch the workflow parameter file appropriate to the workflow containing the actions for which you are enabling time-based escalation.

The following table lists the workflow parameter files that apply to workflows associated with the various COREid applications:

Table 27 Workflow parameter file names and paths

Application	Workflow parameter file name and path
User Manager	<i>Component_install_dir</i> /identity/oblix/apps/userservcenter/bin/usc_wf_params.xml
Group Manager	<i>Component_install_dir</i> /identity/oblix/apps/groupservcenter/bin/gsc_wf_params.xml
Organization Manager	<i>Component_install_dir</i> /identity/oblix/apps/objservcenter/bin/osc_wf_params.xml

- Locate the compound list for the action you want to support time-based escalation. The first half of this compound list should resemble the listing that follows:

Listing 1 The workflow parameter compound list (partial listing only)

```
<CompoundList ListName="actionName">
  <SimpleList >
    <NameValPair ParamName="occurrence" Value="n"/>
    <NameValPair ParamName="useraction" Value="true"/>
    <NameValPair ParamName="initialStep" Value="false"/>
    <NameValPair ParamName="time_based_escalation" Value="true"/>
  </SimpleList>
  . . .
</CompoundList>
```

where *actionName* is the name of the action for which you want to enable time-based escalation. For a list of actions that can be enabled to support time-based escalation, see Table 26, “User Actions Available for Advanced Ticket Routing,” on page 213.

- Add the following string in the position indicated by the preceding listing:

```
<NameValPair ParamName="time_based_escalation" Value="true"/>
```

- Repeat the preceding steps for all the user actions you want to support time-based escalation.
- Save and exit, the file.

Performing Asynchronous Operations

An asynchronous workflow moves from step to step without completing pending subflows. An asynchronous operation is pre and post processing code that is part of an Identity Event, as described in the *NetPoint 7.0 Developer Guide*. The `asynch_user` parameter determines who may resume the pending asynchronous action. The default is Anyone.

To allow a user to perform an asynchronous operation

1. Open the `asynchparams.xml` file in:

```
COREidInstall_dir/apps/asynch/bin/asynch
```

where `COREidInstall_dir` is the directory where you installed the COREid System.

2. Set the `asynch_user` parameter to one of the following:

- **Anyone**—Anyone can perform asynchronous operations (default).
- **DN**—A particular user can perform asynchronous operations. Provide the DN of a user.

Only one DN can be accepted by the parameter.

3. Close the `asynchparams.xml` file.

Notes on Asynchronous Workflows

The User, Group, and Organization Managers are not automatically loaded when an asynchronous workflow is resumed. If there is a request to the COREid Server to resume an asynchronous workflow when the application is not loaded, the workflow engine may not register error conditions.

For example, suppose you create a Deactivate User workflow in the User Manger. This workflow only has Initiate and Disable steps. Suppose also that you create an event plug-in for the workflow that returns `STATUS_PPP_WF_ASYNC` code to make the workflow instance become asynchronous during the Initiate step, pending a command to instruct the workflow to resume and run the Disable step. If there is an IdentityXML request to resume this workflow while the COREid System is being restarted, the workflow engine would mistakenly return success. The Disable step would return with a status of complete when in fact the user was not disabled.

Note: Be sure the User Manager, Group Manager, and Organization Managers are pre-loaded when the COREid Server is restarted.

To preload the User, Group, and Organization Managers

1. Open the COREid parameter file:

```
COREid_install_dir/identity/oblix/engine/obengineparams.xml
```

2. In this file, find the configuration information for the COREid applications:

- `<ValNameList ListName="groupservcenter">`
- `<ValNameList ListName="userservcenter">`
- `<ValNameList ListName="objservcenter">`

3. Change the Dll_Load parameter from 0 to 1 as in following example for Group Manager.

```
<Val NameLi st Li stName="groupservcenter" >  
<NameVal Pai r ParamName="DI I _Name" Val ue="groupservcenter" />  
<NameVal Pai r ParamName="DI I _Di r" Val ue="obl i x/apps/  
groupservcenter/bi n" />  
<NameVal Pai r ParamName="DI I _Load" Val ue="1" />  
<NameVal Pai r ParamName="Work_Di r" Val ue="obl i x/apps/  
groupservcenter/bi n" />
```

Using a Workflow

Once a workflow has been defined, users can invoke the workflow from the associated function in the User Manager, Group Manager, or Organization Manager. Participants in steps other than the Initiate step of the workflow can find and process tickets. Users can delete workflow requests, archive requests, and monitor the progress of a workflow.

Note that to be able to perform the actions specified in the workflow definition, participants in a workflow must be granted permission to view and modify the attributes affected by the workflow. See “Allowing Users to View and Change LDAP Data” on page 126 for details.

Invoking a Workflow

Once a workflow is defined, it becomes a piece of embedded functionality in the User, Group, or Organization Manager. The workflow can be invoked by any user who has been defined as a participant in the Initiate step for this workflow. For example, suppose you define a Create User workflow. Users who are in the domain specified for the workflow can invoke this workflow from the Create User function in the User manager. If multiple workflows have been defined for a create operation, you will see a drop-down list on the create page for that object.

Workflow Name : Create User with LDAP filter **Workflow Type :** Create User

Defined Steps: [step 1] Initiate

[New] [Modify] [Delete Step] [Insert Step]

Step Properties

Action Subflows Attributes Participants Mail Notification

Select Participants*

Select Role: [Anyone]

Build Filter: [Build Filter]

ldap://OU=Company,DC=qalab-fishcake,DC=oblix,DC=com??sub?(genbadg)

Select Person(s): [Select User]

<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Select Group(s): [Select Group]

[Save Step]

[Save Workflow] [Cancel Workflow] [Previous]

Users can also initiate a Change Attribute workflow. Change attribute workflows are available on profile pages that the user is permitted to access. For example, suppose a workflow has been defined for the manager attribute displayed on a profile page. When users change departments, they may need to issue a request to change the name of their manager. This request can be handled by a Change Attribute workflow.

To invoke a change user attribute workflow

1. From the User Manager, click My Identity > Modify.

Your user profile page is displayed using editable fields for all attributes that you can change.

2. For attributes on your profile page that have the Request to Remove or Request to Modify buttons, you may request to remove or change that attribute value.

Your request is sent in the form of a ticket. The person who processes this ticket may approve or reject the request. See “Finding and Processing a Ticket” on page 230 for details.

Finding and Processing a Ticket

Once a workflow has been initiated, subsequent steps are generated by processing a ticket. You can find pending workflow tickets in the User Manager, Group Manager, and Organization Manager.

To find a workflow ticket

1. From the User Manager, Group Manager, or Organization Manager, click Requests.
2. From the Requests page, click either Incoming Requests, Outgoing Requests, or Monitor Requests.

Note that outgoing requests are tickets that have already been processed.

3. In the Search drop-down list, select the application for which you want to view requests.
4. Specify a number of days in the text field, or leave this field blank to view all requests.
5. Click Go.

A list of workflow tickets is displayed. The list will match your search criteria.

To process a ticket

1. From the User Manager, Group Manager, or Organization Manager, click Requests.
2. From the Requests page, click Incoming Requests.
3. In the Search drop-down list, select the application for which you want to view requests.
4. Specify a number of days in the text field, or leave this field blank to view all requests.
5. Click Go.
6. A list of workflow tickets is displayed. The list will match your search criteria.
7. Click a link for an incoming request.
8. On the details page for the request, click the Process button.

If you are a participant for this workflow, a page is displayed showing the attributes configured for this step of the workflow.

9. Supply any required attributes for this workflow.

For example, a Create User step may prompt you to supply an email address for the new user. Any information you need to supply on this page is determined by how this step of the workflow was configured.

10. Click the appropriate button for completing this step of the workflow.

For example, on a Create User request, the detail page for this ticket may contain Approve and Reject buttons.

Deactivating and Reactivating Users

Once a user has been enabled in NetPoint, they can be deactivated and reactivated. Deactivation makes a user unable to log in and unavailable for viewing in the COREid System. It takes effect once the user logs out of the current session. An administrator with Monitor Requests privilege can view deactivated users and either permanently delete them or reactivate them.

Note: All configured directories will be searched to remove references to the Deactivated/Deleted user, including groups to which the user belongs. When you have stored user data and NetPoint configuration data separately, both directories will be searched concurrently.

The steps for defining a workflow for deactivating a user are provided in “Using the Workflow Applet” on page 194. Once the workflow has been defined, users with sufficient privileges will see an Initiate User Deactivation button on a user’s profile page.

Initiate User Deactivation View Panels Modify

User Profile

Badge Photo 

Title Manager, IT
Name Lars Usersen

Telecommunications Information

Car License 1
Mobile Phone Number (888) 888-8888
E-Mail Address lusersen@oblix.com

The steps for deactivating the user will conform to the actions you specified on the user deactivation workflow.

Reactivating a Deactivated User

At times you may want to reactivate a deactivated user. For example, you may want to deactivate an employee during a leave of absence, and reactivate the employee when he or she returns to work.

To reactivate a deactivated user

1. Define a Reactivate User workflow for this purpose.

For a summary of actions permitted on a reactivation workflow, see “Workflow Types, Steps, and Actions” on page 179. Once the workflow has been defined, users with sufficient privileges will see an Initiate User Reactivation button on a deactivated user’s profile page.

2. Navigate to the Deactivated User Identity tab to display the Search Deactivated Users page.
3. Search for, then select the deactivated user name for the identity you want to reactivate.

The View Profile page appears.

4. Click the Initiate User Reactivation button on the View Profile page.

The user is reactivated.

Note: When you reactivate a user, you must manually add the user to any groups he or she belonged to, and re-set attribute policies and the searchbase for the user.



Monitoring a Workflow

Users who have the right to monitor a workflow can view the progress of a workflow, including request tickets owned by others.

Only requests within your management domain are listed. See “Delegating Administration” on page 47 for more information.

To monitor a workflow

1. In User, Group, or Organization Manager, click Requests.
2. Click Monitor Requests.

Note: For subflows, if the first step has not been processed, the Date Processed field is empty.

3. In the Search fields, select your search criteria and click Go.

The results appear below the search fields.

4. Click Next or Previous as necessary to see other results.

5. Click a ticket's Request Number to open the Request page for that ticket.

This page lists the workflow's current step number.

To delete an incomplete workflow that is not responding, use the Monitor Requests function to locate the workflow and then click the Terminate button.

To terminate a completed workflow, use the Delete button in the Monitor Workflow functionality.

Archiving Requests

You may want to archive workflows to keep a record of participants and times, and to prevent the Oblix configuration tree from getting too large. Archived workflows are stored in LDIF format. The default storage file is `oblix/data/common/wfinstance.ldif`. Multiple archive operations add information to this file.

You can archive only completed workflows.

To archive a workflow

1. View workflow requests, as described in See "Monitoring a Workflow" on page 233.
2. Select requests in the Select All column.
3. Click Archive.

You can change the default filename and location in the following files:

Filename	Application
<code>usc_wf_params.xml</code>	User Manager
<code>gsc_wf_params.xml</code>	Group Manager
<code>osc_wf_params.xml</code>	Org. Manager

4. When the archive confirmation page appears, click Back to return to the previous page.

Note: You must restart the COREid Server after changing parameter files.

Deleting Requests

You can remove workflow requests.

To delete requests

1. View requests as described in See “Monitoring a Workflow” on page 233.
2. Select requests in the Select All column and click Delete.
3. When the delete confirmation page appears, click Back to return to the previous page.

Preventing Other Administrators from Working on a Workflow Ticket

At run time, multiple users may receive a workflow ticket. For example, suppose an IT group receives a ticket for a Create Workflow request. An administrator who processes this request can lock the ticket so that other users can view the information on the locked ticket but they cannot work on the ticket. Only the person who locked the Ticket, the Master Identity Administrator, and people who have been granted permission to Monitor Requests can unlock the ticket.

To lock or unlock a ticket

1. Open a ticket as described in “To process a ticket” on page 231.
The Lock and Unlock buttons are displayed on the workflow page when you process the workflow ticket.
2. Select Lock or Unlock, as appropriate.

Managing Workflows

Once you have defined workflows, you can view, copy, modify, delete, and export them.

Viewing and Exporting a Workflow Summary

You can view a summary of a workflow, including its steps, participants, and so on, and export this report to a comma-separated value (CSV) file.

Note: If you are using Microsoft Internet Explorer, to enable the Export to CSV feature, two parameters in the WebGateStatic.lst file must have the following settings:

CachePragmaHeader—Leave blank

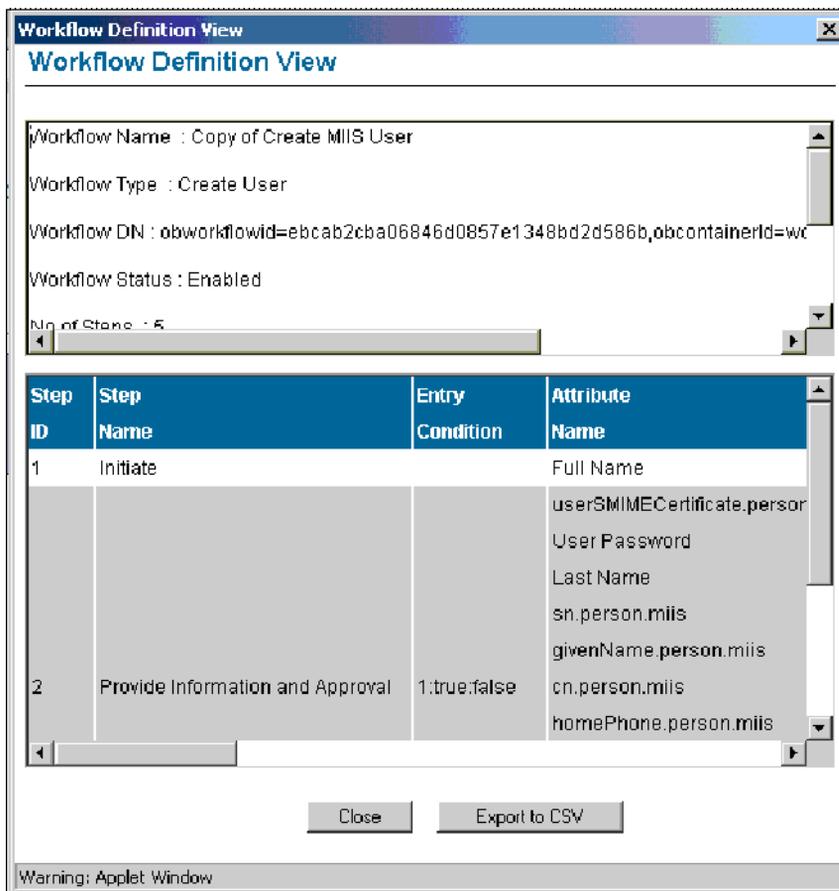
CacheControlHeader—Specify Private or leave blank

This file is of interest to you if you are using the NetPoint Access System, and you are protecting the COREid System interface (WebPass) with a WebGate.

To view and export a workflow summary

1. Access the User, Group, or Organization Manager.
2. Click Configuration > Workflow Definition.
3. From the workflows drop-down menu, select the workflow you want to view.
4. Select View.

The Workflow Definition View page appears.



5. Enlarge the Workflow Definition View page or scroll to the right to see all of the workflow contents.
6. Click Export to CSV to save a comma-separated value file of your workflow.
7. Click Close to close the page.

A sample CSV file, when opened in a spreadsheet, may appear as follows:

	A	B	C	D	E	F	G	H	I	J
1	Workflow Name	Create MIIS User								
2	Workflow Type	Create User								
3	Workflow DN	ob\workflow\id=ebcab2c9a06846d0957e1348bd2d586b_obcontainerId=workflowDefinitions_o=Oblix_o=company_c=us								
4	Workflow Status	Enabled								
5	No. of Steps	5								
6	Description									
7	Workflow Target	MyMIISUser_ou=Ford,o=Company,c=US								
8	Workflow Domain	o=company,c=us:								
9										
10										
11	Step ID	Step Name	Entry Condi	Attribute Name	Attribute Kind	Attribute Proc	Attribute Defa	Selected Sub	Commit Dome	Participant Ro
12	1	Initiate		Full Name	Required					Anyone
				user\$MIMECertf						
				icate.person.mis						
				User Password						
				Last Name	Optional					
				sn.person.mis	Optional					
				givenName.pers	Required					
				on.mis	Optional					
				cn.person.mis	Optional					
				homePhone.pers	Optional					
				on.mis	Optional					
				mail.person.mis	Optional					
				title.person.mis	Optional					
				Login	Required					
13	2	Provide Informatic	1 true false	uid.person.mis	Optional	0	0			Anyone
14	3	Activate	2 true false						LDAP	Anyone
15	4	Activate	3 true false						mis	Anyone
16	5	Error Report	4 false false							Anyone

Copying a Workflow

You can use a copy of a workflow as a starting point for a new workflow.

To copy a workflow

1. Access the User, Group, or Organization Manager.
2. Click Configuration > Workflow Definition.
3. From the Workflows drop-down menu, select the workflow you want to copy.
4. Click Copy.
5. A copy of the workflow appears in the list.

It is named Copy of *original name*. Oblix recommends renaming the copied workflow, although you are not required to do so.

6. Change information as necessary to create a new workflow.

Modifying a Workflow

After creating a workflow, you can change its parameters. If the selected workflow has pending instances, you can only modify the list of Targets, the Participants for any step, or the Pre/Post Notification recipients for any step.

To modify a workflow

1. Access the User, Group, or Organization Manager.
2. Click Configuration > Workflow Definition.
3. From the Workflows drop-down menu, select the workflow you want to modify.
4. Click Modify.

The selected workflow information appears. Clicking Modify disables this workflow so that it cannot be used while being modified.

5. Change the workflow settings as necessary.
6. Click Save Workflow to save your changes.
7. Click Yes when prompted to enable the workflow.

Deleting a Workflow

You can delete a workflow unless the workflow has pending requests.

To delete a workflow

1. Access the User, Group, or Organization Manager.
2. Click Configuration > Workflow Definition.
3. From the Workflows drop-down menu, select the workflow you want to delete.
4. Click Delete.
5. Click OK at the confirmation message.

Exporting Workflows

You can export all workflows to a comma-separated value (CSV) file. This is a text file that can be printed.

Note: If you are using Microsoft Internet Explorer, to enable the Export to CSV feature, these parameters in the WebGateStatic.lst file must have the following settings:

CachePragmaHeader—Leave blank

CacheControlHeader—Specify “Private” or leave blank

To export workflows

1. Access the User, Group, or Organization Manager.
2. Click Configuration > Workflow Definition.
3. From the workflows drop-down menu, select the workflow to export.

4. Click Export All to be prompted to save a comma-separated value file that includes all of your workflows.

Viewing Workflow Panel Settings

As described in “About User, Group, and Organization Manager” on page 100, you configure what appears in the User, Group, and Organization Manager applications. The User and Group Manager applications consist of one tab and the Organization Manager consists of one or more tabs. Tabs are a collection of profile pages, which themselves are collections of panels. Panels are groups of attributes and values.

You can view and modify the workflow panels that appear on the profile pages for these applications.

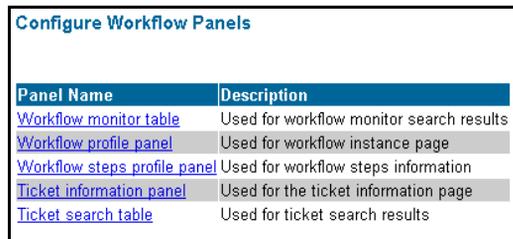
To view current workflow panel settings

1. From the COREid System Console, click Common Configuration.

The Common Configuration page appears.

2. Click Configure Workflow Panels.

The Configure Workflow Panels page displays configured workflow panels.



Panel Name	Description
Workflow monitor table	Used for workflow monitor search results
Workflow profile panel	Used for workflow instance page
Workflow steps profile panel	Used for workflow steps information
Ticket information panel	Used for the ticket information page
Ticket search table	Used for ticket search results

The following table describes each panel.

Panel	Description
Workflow monitor table	The columns included in the results page when a user performs a workflow search from the Monitor Requests page.
Workflow profile panel	The information displayed about a workflow instance in the Monitor Requests page.
Workflow steps profile panel	The information displayed about a workflow instance's steps in the Monitor Request page.
Ticket information panel	The information displayed in the Ticket Information page from the Incoming Requests or Outgoing Requests page.
Ticket search table	The information displayed in the results page when a user performs a workflow search from the Incoming Requests or Outgoing Requests page.

3. Click the panel you want to view.

The View Panel page displays the items that are displayed on the panel.

Panel field	Description
Panel Label	Name of the panel as displayed in NetPoint. This field can be localized.
Description	Description of what this panel does. This field can be localized.
Attributes	Attributes used for the panel columns and their display names. This field can be localized.

Modifying the Appearance of Workflow Panels

You can modify, but cannot delete, workflow panels.

To modify a workflow panel

1. From the COREid System Console, click Common Configuration.

The Common Configuration page appears.

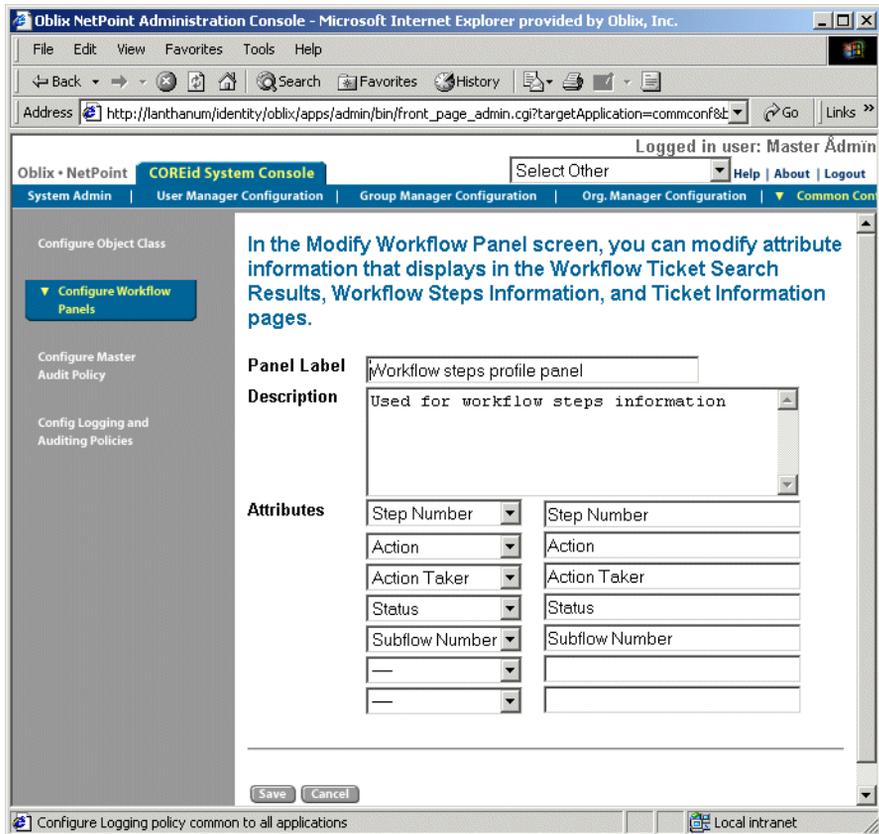
2. Click Configure Workflow Panels.

The Configure Workflow Panels page displays configured workflow panels.

3. Click the panel you want to view.

4. Click Modify.

The Modify Panel page appears.



5. In the Panel Label field, type a new name for this panel as it will appear in the application.
6. In the Description field, type a description.
7. In the Attributes fields, select attributes to display on the application in the order in which they will appear.
8. Click Save.

Localizing Workflow Panels

You can localize workflow panels if you want to display the panel information in more than one language. To do this, you must do the following:

- Install the appropriate language packs.
- Manually enter the panel display information in the Administration Console for each language that you installed.

See “Making Schema Data Available to NetPoint” on page 59 and “Configuring User, Group, and Organization Manager” on page 99 for information on localizing attributes.

To view language-specific workflow panel information

1. From the COREid System Console, click Common Configuration.

The Common Configuration page appears.

2. Click Configure Workflow Panels.

The Configure Workflow Panels page displays configured workflow panels.

3. Click the panel you want to view.

The details of the workflow panel such as the panel name, description, and attributes are displayed on the page.

4. Click Translate.

Note: The Translate button appears only if more than one language pack is installed.

The Summary of Panel Display Names page appears. The language-specific display name for the panel fields and attributes are displayed. Fields that has not been translated for a language are marked as Not Configured.

To configure language-specific workflow panel information

1. In the Summary of Panel Display Names page, click Modify.

The Configure Panel Display Names page appears. This page contains panel information and links for the languages that you have installed

2. Click the language for which you want to configure the workflow panel.

The Configure Panel Display Names page for the selected language appears.

3. Enter the following information:

- **Panel Label**—Enter the language-specific display name for the panel.
- **Description**—Enter a brief description of the panel. This is optional.
- **Attributes**—Enter the language-specific text for each attribute display name.

4. Click Save to save your changes; click Cancel to exit the page without saving your changes.

Workflow Performance

Access to the directory server access can be reduced by setting the WfInstanceNotRequired flag to true in the oblix/data/common/workflowdbparams.xml file. This flag indicates that no workflow instances should be written to the directory server unless necessary. It is set to false by default, which means workflow instances are written to the directory server.

The NetPoint Identity Administrator's Modify Rights

As defined in “Specifying COREid System Administrators” on page 43, only a NetPoint Identity Administrator can manage the User, Group, and Organization Manager.

By default, a NetPoint Identity Administrator bypasses attribute access controls. As a result, if you define a Change Attribute workflow, the attribute access controls in this workflow are not checked for NetPoint Identity Administrators. These administrators automatically have modify rights where other users have only the right to request to modify an attribute.

The parameter to control this feature is `BypassAccessControlForDirAdmin`, located in `COREid/identity/oblix/apps/common/bin/globalparams.xml`. If you want NetPoint to not automatically provide modify rights to the Directory Administrator, set this flag to false and restart the COREid Server.

You can give the NetPoint Identity Administrator the right to modify an attribute and request to modify an attribute in a Change Attribute workflow. In each application parameter file, for instance, `COREid/identity/oblix/apps/userservcenter/bin/userservcenter.xml`, you can set the parameter `checkChangeAttributeEvenAllowModify` to true. This setting provides that even if the administrator is allowed to modify an attribute, this person will see both input and workflow buttons. This parameter applies to administrators who have both modify and initiate workflow rights. Note that this feature can introduce performance overhead.

Advanced Workflow Options

The following advanced options are available:

- Attaching custom code to workflow actions
- Configuring the behavior of workflow actions

Pre and Post Actions

You can use the Identity Event Plug-in API to attach custom code to workflow actions. Common scenarios for using the Identity Event Plug-in API with workflows include:

- Automatically generating a value (such as a unique ID) from an external system
- Validating data for a workflow step
- Updating data in an external system

Once you write custom code, you must tell the COREid System to execute it either before the workflow action (pre action), or after the workflow action (post action) in the `oblixppcatalog` file.

See the *NetPoint 7.0 Developer Guide* for more information.

External Actions

An external action serves the same purpose as pre and post actions, but differ from Identity Event Plug-in API actions in two ways:

- An external action is not attached to an existing action.
- Routing paths can be fully configured based on the external action's exit condition.

You implement the external action code as a hook in the `oblixppcatalog` file. See the *NetPoint 7.0 Developer Guide* for more information.

Customization of Data and Actions in a Workflow

The User, Group, and Organization Manager each have a workflow parameter file that controls the data displayed to participants and the actions that can be selected.

Each parameter file has three sections:

- Create Object
- Delete Object
- Change Attribute

The files are located in

`COREidInstall_dir/identity/oblix/apps/applicationname/bin/`

where `COREidInstall_dir` is the COREid Server installation directory and `applicationname` is one of the following:

- `usc_wf_params.xml`: User Manager
- `gsc_wf_params.xml`: Group Manager
- `osc_wf_params.xml`: Org Manager

The following table describes each parameter:

Parameter	Description	Sample Setting
occurrence	Indicates how many times this action may be used within a workflow.	[1] [n] 1—action can be used once. n—Action can be used multiple times.
useraction	Indicates whether or not the step is interactive.	[true] [false] true—Action requires user interaction. false—This is a background step and requires no user interaction.
forceCommit	Indicates whether an implicit commit takes place for this step, even though this action is not a commit. An implicit commit writes all collected data to the specific target entry.	[true][false] true—Implicit commit takes place. false—Implicit commit does not take place.
pre_action	Indicates that the current action can be specified if the previous step's action is in this list.	[list of actions]
exit_condition	Indicates the possible results for the given action.	[list of exit conditions] For example: true: 1 false: 0
relevant_data	Indicates which types of relevant data can be configured for this step. Background steps do not contains any relevant data.	[list of relevant data] Can be any combination of Required, Provisioned or Optional.
initialStep	A parameter you can apply to an initiate, self registration and/or approval step.	Values are true and false.

Adding Roles to a Workflow

By default, only the role of Anyone is available in a workflow definition applet. To add the roles that have been defined in the directory to the workflow definition applet, you can modify the workflow parameter files for the User Manager, Group Manager, and Organization manager.

The following procedures cause all DN roles to show up in the workflow applet.

To configure a role

1. Open the Modify Attributes applet as described in “Configuring Attributes” on page 81.
2. For a person object class or an auxiliary object class associated with the person object class, select an attribute with a DN data type.
For example, you might select the Manager attribute.
3. Select the Object Selector display type for this attribute.
4. Select the person object class in the Target Object Class list.

The attribute will display as a role in the workflow applet, provided all roles are enabled as described in the following procedure.

To add roles to a workflow definition applet

1. Edit the WF parameter file in the following location:
User Manager—Open usc_wf_params.xml.
Group Manager—Open gsc_wf_params.xml.
Organization Manager—Open osc_wf_params.xml.

2. Go to the section `<CompoundList ListName="Roles">`
3. Find the appropriate Workflow Type.

For example, to modify a create object workflow, you would need to find `<CompoundList ListName="CREATE_OBJECT">`

4. Find the “Participant” or “Notiffee” section in this file.

For example, you could edit the section `<ValNameList ListName="Participant" >`

5. Add the following line:

```
<NameValuePair ParamName="dns" Value="dns"/>
```

Creating a Self-Registration Workflow

Self-registration enables users to add themselves or their organizations to the COREid System directly from a web page. NetPoint does not provide a user interface for self-registration. You must configure a URL that displays a registration form.

When users self-register, they may be prompted to reset their passwords after their initial login attempt. This depends on settings provided for the Change On Reset field, as described in “Configuring Password Policies” on page 320. If more than one user self-registers using the same browser session and the Change On Reset option is chosen, all users after the first are prompted to change their passwords after their first login.

If you want users to be automatically logged in to the COREid System after self-registering, you must set the `SelfRegGeneratesSSOCookie` to true in the `basedbparams.xml` file. See the *NetPoint 7.0 Customization Guide* for details.

The following procedure illustrates a self-registration workflow for Basic authentication.

To create a self-registration workflow

1. From the COREid System Console, select the User, Group, or Organization Manager.

If the Organization Manager has more than one tab, select the appropriate tab.

2. Click Configuration > Workflow Definition.
3. Define a Create User or a Create Organization workflow using self-registration as the first step.
4. Access this workflow and record the workflow’s Distinguished Name and the target domain.

You will add this information to the self-registration URL.

5. Add the self-registration URL to an HTML document as follows:

```
https://domain_name:portnumber/identity/obl ix/apps/userservcenter/bin/
userservcenter.cgi?program=workfl owSel f
Regi strati on&ObWorkfl owName=workfl ow DN&ObDomainName=target domain
```

Variables are as follows:

- ***Domain_name:portnumber***—host system
- ***Workflow DN***—the workflow’s DN
- ***Target_domain***—the target path, without *name*

Note: The ObDomainName *target_domain* needs to be one of the target domains defined in the self-registration workflow. For more information, see “Defining an LDAP Target for Create Object Workflows” on page 199.

For organization self-registration, use this format:

```
https://domain_name:portnumber/identity/oblix/apps/objservcenter/bin/objservcenter.cgi?program=workflowSelfRegistration&tab_id=tab_name&WorkflowName=workflow_DN&ObjectName=target_domain
```

Variables are as follows:

- **Domain_name:portnumber**—host system
- **Tab_name**—the name of the tab
- **Workflow DN**—the workflow's DN
- **Target_domain**—the target path, without name

The URL for self-registration must be to a page that does not require authentication. The self-registration URL is not the usual /identity/oblix/apps/userservcenter/bin/userservcenter.cgi. Typically, when a user accesses the COREid System, the Access System asks the user to authenticate. However, the WebGate should be set up to not request authentication for people accessing self-registration and lost password pages.

6. Replace reserved characters with URL-compatible text substitutes.

When providing a DN path in the dynamic expansion URL, you must encode URL-reserved characters (non-alphanumeric) with a % followed by the character's ASCII hexadecimal equivalent, as follows:

- **%3D**—Equal sign (=)
- **%2C**—Comma
- **%20**—Space

For example:

cn=Engineering Team, ou=Engineering, o=Company, c=US

is replaced by:

cn%3DEngineering%20Team%2C%20ou%3DEngineering%2C%20o%3DCompany%2C%20c%3DUS

7. Save the HTML file.

The following is an example of a self-registration URL:

```
http://silicon/identity/obl/x/apps/userservcenter/bin/
userservcenter.cgi?program=workflowSelfRegistration
&obdomainname=0%3DCompany%2C%3DUS&obworkflowname=
obworkfl owid%3D20020605T1132216476%2CobcontainerId
%3Dworkfl owdefi ni ti ons%2co%3Dobl i x%2Co%3Dconfi gdata
```

Note: If you are using Sun's iPlanet directory, note that self-registration passwords cannot use UTF-8 characters. If the user supplies UTF-8 characters, the iPlanet directory default 7-bit plug-in fails the operation. By default the 7-bit plugin requires the uid, mail, and userpassword attribute values to be 7-bit. To resolve this problem, turn off the plug-in or remove the userpassword attribute from the configuration. Note also that this issue applies as well to Create User and Modify Profile operations.

Creating a Location Workflow

In the Organization Manager, you can create workflows to manage business locations and allow specific users to manage those locations. You can select individual users or users who play a specific role such as Facilities Manager, or you can select specific groups such as IT Operations.

To enable users to view the location of the organization, you can attach .gif images of the location map to the workflow. When users click on a location, the location profile displays a map of the area where building is located.

You can create a new location workflow and then create a location object using the new workflow. To do this, use the Create Org Profile feature in the Organization Manager. Alternatively, you can create a location object first and then link it to an existing workflow. Once you create a location object, you can assign other objects such as users to specific locations on the map.

Note: If Location ID has the Semantic type DN Prefix it is important to note Active Directory and ADAM do not allow multi-valued RDNs (although iPlanet/SunOne do). For Active Directory and ADAM, ensure that the Attribute Value(s) selection is Single in the meta-attribute configuration.

After you have created a location object, you must enable the Location functionality and enable users with the appropriate permissions to view the user or object's location.

Task overview: Enabling Location functionality and users

1. Modify the Location tab for the Organization Manager, then add location attributes to the Profile pages for User Manager and the Organization Manager. See "Enabling the Location Tab in Organization Manager" on page

146 and “Configuring Tab Profile Pages and Panels” on page 113 for more information.

2. Configure read permissions for location attributes. See “Allowing Users to View and Change LDAP Data” on page 126 for more information.
3. Define a Create Location workflow as described in “Task overview: Defining a Create Location workflow” on page 251.
4. Create a new location and establish the location’s hierarchy in relation to other locations if applicable. See “Adding Object Classes” on page 69 for more information.
5. Assign a value for the Location attribute for a user or object profile. See or “Using the Workflow Applet” on page 194 for more information.

Note: Attribute values can also be added and modified on object profile pages as well as through a workflow.

Task overview: Defining a Create Location workflow

1. Initiate a workflow as described in “Starting a New Workflow Definition” on page 196.
2. Create one or more subflows, if necessary, as described in “About Subflows” on page 189.

Note: You must create subflows before you initiate the main workflow. This allows you to link a subflow to the main workflow.

3. Select the attributes that you want to associate with the workflow as described in “Defining Step Attributes” on page 203.

The available default location attributes are Location ID, Location Name, Location Title, and Map Image. Location ID and Location Name are required attributes.

4. Specify participants as described in “Defining the First Step in a Workflow” on page 201.
5. Define the workflow process as described in “About Step Actions” on page 182.
6. Save the workflow.
7. Enable the workflow as described in “Enabling the Workflow” on page 208.
8. Test the workflow to ensure its validity as described in “Testing the Workflow” on page 209.

6 Provisioning External Applications from COREid

The User, Group, and Organization Manager are COREid System applications that enable users to view and modify information about themselves, other people, groups, inventory, and any other item that you, the administrator, choose to make available. As explained in the chapter “Chaining COREid Functions Into Workflows” on page 171, you can apply business logic to actions performed in the COREid applications, so that, for example, review and approval must be performed before information about a user can be modified.

With the COREid System’s provisioning functionality, you can extend a COREid workflow so that information that is added, deleted, or changed is propagated to other applications. This functionality is intended for provisioning user accounts in back-end applications such as email.

This chapter covers the following topics:

- “About Provisioning Application Accounts” on page 254
- “Summary of Provisioning Using a Workflow” on page 254
- “About Template Objects and Provisioning” on page 256
- “About Template Object Data and Workflows” on page 256
- “Object Template Configuration” on page 257
- “Sample Object Template File” on page 262

About Provisioning Application Accounts

The User Manager, Group Manager, and Organization Manager applications rely on information in an LDAP directory or in an object template:

- **LDAP Directory**—You configure the information in the directory to display data on profile pages and to enable configuration of workflows that manipulate data about users, groups, and objects. The LDAP directory is the authoritative data source for the COREid System.
- **Object Template**—You can manually configure information in an object template for the purpose of propagating data that is entered during a COREid workflow step to different target data sources. A back-end application uses this data for provisioning user accounts. For example, you can configure an Add or Modify User workflow to provision user email accounts. Use of an object template is reserved for custom integrations.

Note: NetPoint generates one object template when you install the NetPoint Management Agent for Microsoft Identity Integration Server (MIIS). This template enables COREid to send data to the MIIS product, which in turn provisions user accounts in various target applications. See the *NetPoint Integration Guide* for details.

Summary of Provisioning Using a Workflow

Oblix provides one out-of-box solution for provisioning, using MIIS. See the *NetPoint Integration Guide* for details. For provisioning to any other back-end system, the process is as follows:

Task overview: Configuring a provisioning solution for a back-end application

1. Configure an object template, as described in “Object Template Configuration” on page 257.

The template should contain objects and attributes that can be understood by the back-end application.

2. Store this file in:

`COREid_install_dir\oblix\config\template\xxx.tpl`

where `COREid_install_dir` is the directory where the COREid System is installed and `xxx` is the name of the .tpl file.

3. Configure the template objects and attributes in the COREid System Console, as described in “Making Schema Data Available to NetPoint” on page 59 and note the following:

- Users can only specify values for template attributes in the context of a workflow step.
- Template attributes are not searchable in the COREid System.
- You cannot set View or Modify permissions for template attributes.

You configure access control for template attributes by defining workflow step participants. Only individuals who are step participants have access to these attributes.

- Template attributes cannot be configured as derived attributes.

Note: After you configure template objects and attributes in the COREid System Console, do *not* modify the template file. This restriction is the same as for an LDAP schema, which you also should not change after configuration. Such changes can cause unpredictable behavior and are not supported.

4. Associate one or more template attributes with panels on a tab in a COREid application, for example, the User Manager, as described in “Configuring User, Group, and Organization Manager” on page 99.
5. Create a workflow and associate the template attribute with one or more workflow steps, as described in this chapter and “Chaining COREid Functions Into Workflows” on page 171.

The attribute display name appears on the profile page associated with the workflow. However, the attribute value is not shown. This is because in the current implementation of provisioning, the data flow is only one-way from COREid to the target application. (In a future release, the data flow will be two-way, which will permit the display of these attribute values.) See “About Template Object Data and Workflows” on page 256 for details.

6. Ensure that the workflow has separate enable, commit, and other steps to write the data to each schema that is used in the workflow, as described in “Configuring User, Group, and Organization Manager” on page 99.
7. Configure an external action using the Identity Event API and IdentityXML to send the data in the object template to the back-end application, as described in the *NetPoint 7.0 Developers Guide*.

Note: For provisioned attribute values, the IdentityXML actions of Add and Replace do not apply. The action Replace All is the only action that is used. If you create an IdentityXML statement using Add or Replace, the statement is processed as if you used Replace All.

The rest of this chapter discusses how object templates work in the context of provisioning, and how to configure an object template.

About Template Objects and Provisioning

NetPoint provides a generic object template schema file, located in:

```
COREid_install_dir\oblix\config\template\
```

This is the required location for this file and for any other object template file that you configure.

Template objects created in this file are similar to LDAP objects. The primary difference is that you use LDAP objects and attributes to display data on user profile pages and to configure workflows, whereas template objects are only used in workflows.

You configure a workflow that contains one or more steps that perform actions on the template attributes. When a user invokes the workflow, the COREid System formats the data entered during the relevant step according to the requirements of the object template schema.

The workflow temporarily stores the template attribute values in the step instance. Once the commit step is performed, the data is written to the target back-end system. The data flow is one-way, so that the COREid System does not continue to store this data once it has been written to the back-end system.

Finally, you must create an Identity Event plug-in to send object template data to the target back-end system.

About Template Object Data and Workflows

As described in the previous discussion, you can configure a workflow step that uses template attributes to provision user accounts in back-end applications. Attribute values that the user enters as part of a workflow step, once committed, cannot be displayed on a profile page. This is because the COREid System sends data to, but does not retrieve the data from, the target application.

When you configure a workflow that uses an object template, you may want to configure commit steps that write both object template data and LDAP data. This would permit you to use your directory to display the LDAP attribute value on a profile page, and to use the template attribute value to send the data to the back-end application.

Since the flow of data from COREid to the back-end application is one-way, the user will not be able to verify provisioned data from the COREid System. The user will need to view the target application or its logs to view the data created in the application from the workflow.

If there is an error writing the data to the back-end system, an Identity Event API plug-in can send a message back to the COREid user.

Note: You cannot commit data for all data sources in one workflow step. You must configure one commit step per domain.

Object Template Configuration

An object template file contains schema-like definitions for objects and attributes in XML format. The objects and attributes that you configure in an object template file correspond to values that can be understood by a back-end application to which you want to write data.

All object template files must reside in:

COREid_install_dir/oblix/config/templates

where *COREid_install_dir* is the directory where the COREid System is installed. The file extension for any object template is *.tpl*.

The COREid Server reads the template file upon startup. If your installation uses multiple servers, you must copy the same template files to each server.

You can define multiple template object classes in a single file or in multiple files. If you create multiple files within the same domain, be aware that the COREid System enforces uniqueness of attributes and classes across files. If an attribute or a class already exists within a domain, the template file is not registered when the COREid Server starts up, and the objects cannot be shown in the System Console.

Similarly, the COREid Server cannot register the template file if it contains any syntax error. Instead, a log entry is generated.

Format of the Object Template File

The template file begins with a schema domain statement. The schema domain resolves ambiguity between object classes that have the same names but are used by different data sources, for example, the user object in LDAP and the user object in Microsoft Identity Integration Server (MIIS).

The following is an example of a schema domain statement:

```
ObTemplateDefinition domain="exchange" version="1.0" />
```

At startup time, the COREid Server reads the domain statement. For display purposes, template objects and attributes are shown in the COREid System Console in the following format:

attribute.class.domain

where the *domain* name is taken from the domain statement in the .tpl file.

Note that all domain statements must be unique. If the COREid System detects a non-unique domain, it fails to read the entire .tpl file. Note that the following domain names are reserved and cannot be used in your .tpl file:

- MIIS
- LDAP

The object template file allows you to define arbitrary name/value pairs. These name/value pairs must match those understood by the target application.

The template definition file is in XML format. The following is an example based on the NetPoint connector for MIIS described in the *NetPoint Integration Guide*:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ObTemplateDefinition domain="MIIS" version="1.0">

  <!-- ObAttributeDefinition -->
  <ObAttributeDefinition name="cn" syntax="OB_CIS" maxlen="20" />
  </ObAttributeDefinition>
  <ObAttributeDefinition name="sn" syntax="OB_CIS" maxlen="20" />
  <ObAttributeDefinition name="mail" syntax="OB_CIS" maxlen="20" />
  <ObAttributeDefinition name="phone" syntax="OB_CIS" maxlen="20" />

  <!-- ObClassDefinition -->
  <ObClassDefinition name="User">
    <ObAttributeReference name="cn" required="true">
    </ObAttributeReference>
    <ObAttributeReference name="sn" required="false">
    </ObAttributeReference>
    <ObAttributeReference name="mail" required="false">
    </ObAttributeReference>
    <ObAttributeReference name="phone" required="false">
    </ObAttributeReference>
  </ObClassDefinition>

  <ObClassDefinition name="Group">
    <ObAttributeReference name="cn" required="true" />
    <ObAttributeReference name="sn" required="false" />
    <ObAttributeReference name="uniqueMember"
  required="false" />
  </ObClassDefinition>
</ObTemplateDefinition>
```

How Template Objects Appear in the COREid System

Objects and attributes defined in the object template file appear as follows in the COREid System:

- Each object that you define in the .tpl file becomes selectable from the page displayed when you select the following from the COREid System Console:

Common Configuration > Configure Object Class > Add.

The name of the object class as displayed in the COREid System Console is in the format *class.domain*. The class is taken from the definition that you provide in the .tpl file.

- Each attribute that you associate with an object in the .tpl file becomes selectable from the page displayed when you select Common Configuration > Configure Object Class > *object class link* > Modify Attributes.

The name of the attribute as displayed in the COREid System Console is in the format *attribute.class.domain*. The *attribute* name is taken from the definition that you provide in the .tpl file.

- In each attribute statement, the syntax element determines the attribute data type that is displayed in Common Configuration > Configure Object Class > *object class link* > Modify Attributes.

You cannot choose the data type for a template object attribute from the COREid System Console. You must configure the data type in the syntax element of the attribute definition.

- Whether this attribute is single- or multi-valued, as seen in Common Configuration > Configure Object Class > *object class link* > Modify Attributes, is determined by the attribute definition in the .tpl file.

- Other characteristics of the attribute, such as the display name and semantic type, are configured from the COREid System Console.

The COREid System enforces the use of only one semantic type per domain. For example, you can only assign the Login and Password semantic types once each per domain.

- Unlike LDAP attributes, the attributes you configure in the .tpl file are not searchable and cannot be configured as derived attributes.

Elements in an Object Template File

The elements of the object template file are as follows.

The object template file begins with a list of attribute definitions. These attributes are referenced in the object definitions later in the file:

Table 28 Elements of ObAttributeDefinition

Element Name	Description
name	<p>Name of the attribute. This corresponds to the attribute name displayed in the COREid System Console.</p> <p>This parameter is required.</p> <p>Length: 32 (max.)</p> <p>Format: [(a-z) (A-Z)][(a-z) (A-Z) (0-9)]</p>
syntax	<p>This is the attribute syntax. It corresponds to the attribute Data Type in the COREid System Console.</p> <p>This parameter is required.</p> <p>Format:</p> <ul style="list-style-type: none"> • OB_DN—LDAP DN. This is synonymous with an LDAP DN attribute. This parameter allows you to configure an Object Selector display type in the COREid System Console. This parameter corresponds to an attribute data type of distinguished name in the COREid System Console. • OB_BIN—A binary. This corresponds to an attribute data type of binary in the COREid System Console. • OB_CES—Case Exact String. This corresponds to an attribute data type of string (case-sensitive) in the COREid System Console. • OB_CIS—Case Insensitive String. This corresponds to an attribute data type of string (case-insensitive) in the COREid System Console. • OB_INT—Integer. This corresponds to an attribute data type of integer in the COREid System Console. • OB_TEL—This corresponds to an attribute data type of telephone in the COREid System Console. • OB_POSTAL_ADDRESS—This corresponds to an attribute data type of postal address in the COREid System Console.
cardinality	<p>The cardinality may be single or multi-valued. This corresponds to an attribute value of single or multi in the COREid System Console. If you set this value to multi, you can re-set it to single in the COREid System Console. However, if you set it to single in the .tpl file, you cannot reset it in the System Console.</p> <p>This parameter is optional.</p> <p>Default: <i>multi</i> unless specified otherwise.</p> <p>Format: [single multi]</p>

Table 28 Elements of ObAttributeDefinition

Element Name	Description
maxlen	<p>The maximum data length for the attribute value.</p> <p>This parameter is optional.</p> <p>Default: -1 unless specified otherwise. This setting indicates that no maximum length is enforced.</p> <p>Format: -1 or 1 - <i>n</i> where <i>n</i> is an integer that represents a reasonable maximum length.</p>

Examples:

```
<ObAttributeDefinition name="c" syntax="OB_CIS" cardinality="single" />
<ObAttributeDefinition name="cn" syntax="OB_CIS" cardinality="single" />
```

In the .tpl file, below the list of attribute definitions, the file contains a list of object classes. Each object class contains an ObClassDefinition statement, followed by a list of ObAttributeReference statements.

Table 29 Elements of ObClassDefinition

Element Name	Description
name	<p>Name of the class. This must be a unique name within the domain.</p> <p>This parameter is required.</p> <p>Length: 32 (max.)</p> <p>Format: [(a-z) (A-Z)][(a-z) (A-Z) (0-9)]</p>

Examples:

```
<ObClassDefinition name="User">
<ObClassDefinition name="Group">
```

You associate an attribute with an object by including the attribute in an AttributeReference statement in the object class definition in the .tpl file. Each ObAttributeReference statement must refer to an attribute defined in an ObAttributeDefinition statement:

Table 30 Elements of ObAttributeReference

Element name	Description
name	<p>Name of the template attribute.</p> <p>This parameter is required.</p> <p>The attribute reference must be unique within the object class. The 'name' must be the name of an existing attribute definition (ObAttributeDefinition) in this domain.</p> <p>Length: 32 (max.)</p> <p>Format: [(a-z) (A-Z)][(a-z) (A-Z) (0-9)]</p>
required	<p>Specifies whether the attribute is required or optional in the context of the class definition.</p> <p>This parameter is optional.</p> <p>Default: 'false'.</p> <p>Format: ['true' 'false']</p>

Examples:

```
<ObAttributeReference name="cn" required="true">
<ObAttributeReference name="mail" required="false">
```

Sample Object Template File

The following is an example of an object template file:

```
<?xml version="1.0" encoding="iso-8859-1"?>

<ObTemplateDefinition domain="myapplication" xmlns:dsml="http://www.dsml.org/DSML"
xmlns:obl ix="http://www.obl ix.com/">
  <ObAttributeDefinition name="c" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="cn" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="department" syntax="OB_CIS" cardinality="single"/>
  <ObAttributeDefinition name="l" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="location" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="mail" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="ou" syntax="OB_CIS" cardinality="single" />
  <ObAttributeDefinition name="uid" syntax="OB_CIS" cardinality="single" />
```

```
<ObClassDefinition name="person">
  <ObAttributeReference name="c" required="false" />
  <ObAttributeReference name="cn" required="false" />
  <ObAttributeReference name="department" required="false" />
  <ObAttributeReference name="l" required="false" />
  <ObAttributeReference name="location" required="false" />
  <ObAttributeReference name="mail" required="false" />
  <ObAttributeReference name="ou" required="false" />
  <ObAttributeReference name="uid" required="false" />
<ObClassDefinition name="organizationalUnit">
  <ObAttributeReference name="l" required="false" />
  <ObAttributeReference name="ou" required="false" />
</ObClassDefinition>
<ObClassDefinition name="locality">
  <ObAttributeReference name="l" required="false" />
</ObClassDefinition>
<ObClassDefinition name="country">
  <ObAttributeReference name="c" required="false" />
</ObClassDefinition>
<ObClassDefinition name="computer">
  <ObAttributeReference name="cn" required="false" />
  <ObAttributeReference name="l" required="false" />
  <ObAttributeReference name="location" required="false" />
  <ObAttributeReference name="ou" required="false" />
</ObClassDefinition>
<ObClassDefinition name="group">
  <ObAttributeReference name="cn" required="false" />
  <ObAttributeReference name="mail" required="false" />
  <ObAttributeReference name="ou" required="false" />
  <ObAttributeReference name="uid" required="false" />
</ObClassDefinition>
<ObClassDefinition name="role">
  <ObAttributeReference name="l" required="false" />
  <ObAttributeReference name="ou" required="false" />
</ObClassDefinition>
</ObTemplateDefinition>
```

Creating an Identity Event Plug-In for Template Attributes

The *NetPoint 7.0 Developers Guide* describes how to create a plug-in to send data from a COREid workflow to a back-end application. Keep in mind the following when creating this plug-in:

- It is no longer possible to do bulk reactivations using the Identity Event API.
- When using IdentityXML and the Identity Event API, note that the only permitted action when sending attributes from COREid is Replace All. If you create an IdentityXML statement using Add or Replace, the statement is processed as if you used Replace All.

7

Configuring and Managing the COREid System

This chapter covers a broad range of tasks to help you manage your data, enhance performance, and control the appearance and functionality of COREid applications. For example, through the searchbase and stylesheet, you may want to control what users can view or the actions they can perform in COREid applications. You may want to add additional COREid Servers or WebPasses for better performance.

To help you manage these tasks, you can specify other NetPoint Administrators and Master Identity Administrators, as described in “Specifying COREid System Administrators” on page 43.

This chapter contains the following topics:

- “Configuring Styles for COREid Applications” on page 266
- “Configuring Multiple Languages for NetPoint” on page 272
- “Configuring COREid Server Settings” on page 276
- “Managing COREid Servers” on page 284
- “Managing Directory Server Profiles” on page 290
- “Managing RDBMS Profiles” on page 305
- “Configuring WebPass” on page 310
- “Configuring Password Policies” on page 320
- “Configuring the Access Server SDK for the COREid System” on page 333

Note: You must be a NetPoint Administrator to configure the COREid System. Most tasks in this chapter are performed through the COREid System Console.

See also, “Section III: Performing Common Administrative Tasks” on page 335, which includes details about NetPoint transport security as well as logging, auditing, and monitoring with NetPoint. Also included there you will find details about implementing Microsoft .NET features with NetPoint.

Configuring Styles for COREid Applications

You use styles to change the appearance or limit functionality across COREid applications. A style is a named collection of stylesheets, graphics files, and scripts that define a certain user interface for the system. A style is based on stylesheets that define the appearance of elements in application pages, including the names of fields and functions, the GIF images used to specify the colors, shapes, and sizes of tabs and buttons, and the fonts used for tab and button names.

Styles can be cosmetic or functional. A cosmetic style determines the appearance of COREid applications, such as color, or the appearance of tabs. A functional style determines the functionality of COREid applications. That is, you can add, modify, or remove particular functions on a COREid application page. For example, you can remove the Substitute Rights function from all three COREid applications. NetPoint provides a default style named Classic Style but you can use *PresentationXML* to develop other styles to change the appearance of NetPoint.

You can use the Customize Styles option in the COREid System Console to set the default style, create styles, modify styles, or delete styles. However, when you create or modify a style through the COREid System Console, the system copies the existing stylesheet and renames it. You must then open the stylesheet and manually make the necessary changes.

See the information on designing the GUI with PresentationXML in the *NetPoint 7.0 Customization Guide* for details about creating and modifying stylesheets.

Note: You can change only the appearance of COREID application pages. The System Console uses the NetPoint Style, and this setting cannot be modified.

See the discussions below for more information:

- “Viewing a Style” on page 267
- “Adding a Custom Style Directory” on page 267
- “Deploying a Style” on page 270
- “Changing a Style Name” on page 271
- “Modifying a Style” on page 271
- “Deleting a Style” on page 271
- “Setting the Default Style” on page 272

Viewing a Style

You can use the procedure below to view currently configured styles. This leads to the Customize Styles page, which is the starting point for style-related procedures.

To view currently configured styles

1. In the COREid System Console, click System Admin > System Configuration.

The System Configuration page appears.

2. In the side navigation bar, click Configure Styles.

The Customize Styles page appears. This example shows the Classic Style, the default provided by NetPoint.



3. Click the style's link to view a style's parameters.

You will see the style name, directory where style files are stored, and the source of those files (in the Copy from field).

Adding a Custom Style Directory

NetPoint out-of-the-box provides one default presentation style, known as Classic Style. The `COREid_install_dir/identity/oblix/lang/en-us/style0` directory contains XSL wrapper stylesheet files for the NetPoint Classic Style. Most of these files point to global shared stylesheet template files for all languages in the `COREid_install_dir/identity/oblix/lang/shared` directory.

The process of creating a custom style for presentation of NetPoint pages for user applications begins by adding a new style to NetPoint, as described here. The result is a new custom style directory with XSL wrapper stylesheet files. Then you may either copy and modify an existing style or create an entirely new style based on new stylesheets.

Note: You may only change styles for user applications. The System Admin Console always uses the default style.

In either case, adding a style (and custom style directory) to NetPoint follows the same method: you provide a style name and a directory name for your style files. You may also choose an existing style on which to base your new style. After selecting your new style as the default, you can begin customizing copies of NetPoint stylesheets or creating your own. To complete the process, you need to

copy your new stylesheets and GIFs to all COREid Servers and WebPass machines, respectively.

Before adding your first new style, there are a few things to take into account:

Multiple Languages—To support multiple languages, NetPoint provides a specific named directory for each installed language. For example, /lang/en-us is the default English language directory, /lang/fr-fr is the French language, and so on. Both the NetPoint default and your custom style directories are stored within each installed language directory.

Suppose you have a French Language Pack installed. In this case, both lang/en-us and lang/fr-fr directories include the /style0 directory. When you add a style to NetPoint, your new style directory is added in both the lang/en-us and lang/fr-fr directories:

```
COREid_install_dir/identity/oblix/lang/en-us/NewStyle
COREid_install_dir/identity/oblix/lang/en-us/style0

COREid_install_dir/identity/oblix/lang/fr-fr/NewStyle
COREid_install_dir/identity/oblix/lang/fr-fr/style0
```

Your Style Name—NetPoint uses the style name you supply internally. As a best practice, your style name should match your custom style directory name and should be easily recognizable. Do *not* include white spaces, &, *, or parentheses () in the name.

Your Style Directory Name—The directory name you specify will be used to create a directory for related wrapper stylesheet files. This name should match your style name and follows the same rules for naming.

In addition, your custom style directory name is also assigned to an XML document (a duplicate of style0.xml) that is created to identify the status and origin of your new style. For example, if your new directory is named *Pastel*, a file is created and stored as:

```
COREid_install_dir/identity/oblix/config/style/Pastel.xml.
```

No other files are created during this process. However, the styles.xml file will be updated to include a NameValPair specifying the directory and style name and directory name that you supply. For example:

```
COREid_install_dir/identity/oblix/config/style/styles.xml
```

The style information files in the config/style are not included on WebPass. For more information, see the *NetPoint 7.0 Customization Guide*.

Copy from an Existing Style—You may copy stylesheets from an existing style directory or select None to build a style that is *not* based on an existing style or to customize only selected stylesheets.

Note: If this is the first style being added, the only available style is the NetPoint default, Classic Style.

Selecting None—If you select None, the directory that is created is empty and you must manually create a set of stylesheets for your new style or selectively copy files from the /style0 directory to work with.

If you select None, your new style’s status will appear as “Under Construction” within NetPoint until you select a style for it. An empty style directory is created automatically, and a duplicate of style0.xml is created in `COREid_install_dir/identity/oblix/config/style/style0.xml`.

Selecting a Style—When you select a style to copy from, a duplicate of the directory you copied from is created under the custom directory name you specify. The copied files retain relative references to the directory you copied from (/style0 or a custom style that you chose to copy from).

During customization, you only change references that refer to the changed version of the stylesheet in the your new style directory.

Results—Suppose you added a new style named *Pastel* in a directory named *Pastel* and you copied from the default Classic Style. In this case, the *Pastel* directory is created in each *langTag* directory and populated with duplicate files from Classic Style’s directory, /style0:

```
COREid_install_dir/identity/oblix/lang/en-us/Pastel
```

The Classic Style directory, /style0, remains intact as:

```
COREid_install_dir/identity/oblix/lang/en-us/style0
```

In addition, an XML document that *duplicates* style0.xml is created when the new style is selected as the default, named after the directory you created, and stored with style0.xml in config/style:

```
COREid_install_dir/identity/oblix/config/style/Pastel.xml  
COREid_install_dir/identity/oblix/config/style/Pastel.xml.lck
```

For additional information and a look at the content of various files, see the *NetPoint Customziation Guide*.

To add a style

1. In the COREid System Console, click System Admin > System Configuration > Configure Styles.

The Customize Styles page appears.

2. Click the Add Style button to display the Add Style page.

Note: The style name you specify here is used internally by NetPoint and should match the directory name you supply.

3. Fill in the fields on the Add Style page, as indicated below:

Name—*Pastel*

Directory Name—*Pastel*

4. In the Copy From field, select an existing style to use as a template for your new style.

For example:

Classic Style

5. Click Save to save the new style (or Cancel to exit this page without saving the style).

The new style name is listed in the Customize Styles page and one or more directories were created to hold the new wrapper stylesheets.

6. Select the new style as your default style, as follows:

- a) Click the Setup Default Style button to display the Set Default Style page.
- b) Click the Make Default button beside your new style name, then click Save.

7. Check your file system for the new style directory name you specified.

Next, you will customize styles, as discussed in the *NetPoint 7.0 Customization Guide*.

Deploying a Style

The following procedure allows you to make a style available to end users.

To deploy a style

1. Append the directory name containing the stylesheets to the URL of the first page where users enter the COREid System, as follows:

&style=directory_name

where *directory_name* is the name of the directory that contains the stylesheets for the COREid System.

Changing a Style Name

You can change the name of a style using the steps below as a guide.

To change a style name

1. In the Customize Styles page, click the name of the style that you want to rename.

The View Style page appears.

2. Click Modify.

The Modify Style page appears.

3. Change the style name.

4. Click Save to save your changes (or Cancel to quit without saving your changes).

The View Style page displays the style's new name.

Modifying a Style

You *cannot* modify Classic style, the default style provided by NetPoint, because it is used by the NetPoint COREid System. However, you *can* modify any of the custom styles you have created.

To change a style

1. Modify the corresponding stylesheets, as discussed in the chapter “Designing the GUI with PresentationXML” in the *NetPoint 7.0 Customization Guide*.
2. When you have completed your changes, copy the stylesheets to each COREid Server and to each WebPass linked to the COREid Server where the stylesheets were modified.

Deleting a Style

You cannot delete the Classic Style, the default style provided by NetPoint, because it is used by the NetPoint COREid System. However, you can delete any of the custom styles you have created.

To delete a custom style

1. In the Customize Styles page, click the name of a style.

The View Style page appears.

2. Click Delete.

3. When prompted, confirm your deletion.

The Customize Styles page reappears.

4. Delete the new stylesheets from the new style directory in all the other COREid Servers, as well as the WebPass installation area.

Setting the Default Style

You use the Setup Default Style option to choose the default style for applications.

Note: You *cannot* select a style that has the *Under Construction* status.

To set the default style

1. In the Customize Styles page, click Setup Default style.

The Set Default Style page appears.

2. Click Make Default beside your choice.
3. Click Save.

In the Customize Styles page, the words “Available & Current Default” appear next to the style you selected.

Configuring Multiple Languages for NetPoint

NetPoint provides the capability to localize NetPoint applications for end users. NetPoint applications include the three COREid applications and the Access Manager.

Within a COREid Server installation, you can install Language Packs such as French and German. This enables you to display static data such as error messages and display names for tabs, panels and attributes to users in their native language.

Task overview: Using NetPoint multi-language functionality

1. Install and set up the COREid Server, as described in the *NetPoint 7.0 Installation Guide*.

During setup, a language entry is automatically created in the Oblix tree for the language in which NetPoint is installed. English (en-us) is the default language.

2. Install language packs of your choice, as described in the *NetPoint 7.0 Installation Guide*.

You can install multiple Language Packs to meet the needs of your users. The Language Pack installer automatically performs the following tasks:

- a) Creates a *langTag* folder under /oblix/lang in the installation directory. A *langTag* folder is where a specific language is installed.

- b) Creates a language entry for each installed language under the Oblix node as follows: *obid=langTag, configDN*.

configDN is the configuration DN in the LDAP directory.

- c) Updates the *obnls.lst* configuration file when you install a language.

The *obnls.lst* configuration file is located in *COREid_install_dir/identity/oblix/config* directory. *COREid_install_dir* is the directory where COREid is installed.

3. Configure the COREid applications for an installed language.

You do this by manually entering the display names for labels and attributes in the COREid System Console. You can localize attribute display names at the following levels:

- Object Class level
- Tab level
- Panel level
- Search Result Attributes level

Note: It is recommended that you configure display names at the Object Class level because this is the highest level. If you choose to configure display names at a lower level, ensure that you provide display names at each level for all languages.

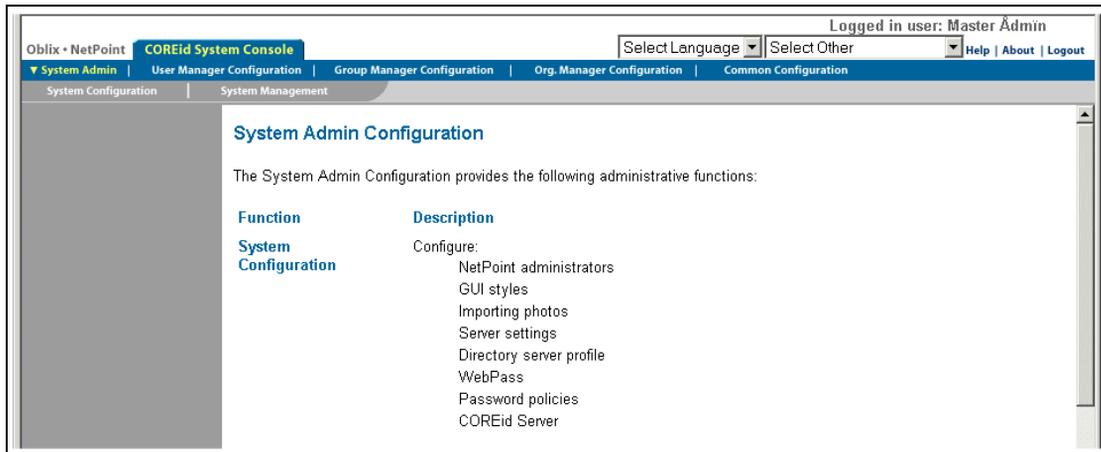
Some display names and attributes appear in the COREid applications as well as the Access Manager. These display names can be viewed in the Access Manager when you configure them in the COREid Server.

For information on:

- Localizing object class attributes, see “Localizing Attribute Display Names” on page 86.
- Localizing display names for tabs, group type panels, search result attributes, and reports, see “Configuring User, Group, and Organization Manager” on page 99.
- Localizing workflow panel names, see “Localizing Workflow Panels” on page 242.

Selecting a Language to Display

If you have installed and configured more than one language, you can determine what language to display on the NetPoint COREid and Access System Consoles. The Select Language drop-down list is displayed at the top of the COREid System Console page. From the drop-down list, you can choose a different language in which to display the NetPoint Administration Console and NetPoint applications.



When a language is selected from the drop-down list, then all requests during the user's session are displayed in the specified language until the session is closed or a different language is specified in the URL. The language preference is stored in a parameter named LangCookie that is embedded in the ObTEMP cookie.

When a user specifies a language, NetPoint searches for the Language Pack in a specific order. The order is as follows:

- NetPoint looks for a Language Pack that is an exact match.
- If an exact match is not found, NetPoint looks for a partial match based on language code.
- If a partial match is not found, NetPoint displays a "Not Configured" message in the display field.

The following example illustrates the order in which NetPoint determines the language. If the following languages are installed:

- en-us
- en-gb
- fr-fr

Then the language in which contents are displayed are as described in Table 31:

Table 31 Order of Installed Languages and Messages Displayed

Order of Installed Languages	Language/Message Displayed	Comments
en-us	en-us	Exact match
en-gb	en-gb	Exact match
en-ca	en-us	Partial match based on the “en” language code
de-de	“Not Configured” message	No exact or partial match found
fr-ca	fr-fr	Partially matched in order

Language Evaluation Order for NetPoint Applications

NetPoint determines the language in which COREid applications are displayed to a user in the order below.

Evaluation Order

1. The language specified in the URL.

The user can specify a language in a URL. For instance, when a user selects the Create User function in the User Manager, he or she can append `lang=fr-fr` to display the User Manager page in French. NetPoint first looks for a language preference specified in the URL for a resource. The user or the administrator can specify a language in the URL by appending `lang=languageTag`, where *languageTag* is a language tag in RFC 1766 format.

The following example returns the Create User Profile page in French:

```
http://localhost/identity/oblinox/apps/userservcenter/bin/userservcenter.cgi?program=workflowCreateProfile&tab_id=employees&lang=fr-fr
```

2. The language stored in a parameter called LangCookie in the ObTEMC cookie.

Once you specify the language on a URL as in the step above, this language is set in the LangCookie parameter. The ObTEMC cookie is created when the user logs in and is maintained for the duration of the user’s session. If a URL does not contain the language specification, NetPoint checks the ObTEMC cookie, which lasts for the duration of the session. The ObTEMC cookie can also be set on a form or a page.

3. The language specified in the HTTP header variable, HTTP_OBLIX_LANG.

You can create an authentication or authorization success header variable to contain this value, as explained in the chapters on authentication and authorization in this manual. If you want to change the name of the HTTP_OBLIX_LANG header variable, you can do so in the following files:

```
COREid_install_dir/oblix/apps/common/bin/globalparams.xml
AccessManager_install_dir/access/oblix/apps/common/bin/
globalparams.lst
```

where *COREid_install_dir* is the directory where COREid is installed, and *AccessManager_install_dir* is where the Access Manager is installed.

See the *NetPoint 7.0 Customization Guide* for information on globalparams.xml.

4. The value set by the user's Web browser to determine the default language. This value is specified in the header variable, Accept-Language.

If NetPoint does not find the HTTP_OBLIX_LANG header variable, it looks for the Accept-Language header variable that is set in the user's browser.

Note: Both the HTTP_OBLIX_LANG and the Accept-Language header variables are configurable. See the *NetPoint 7.0 Customization Guide* for information.

5. The default language of the NetPoint installation.

If NetPoint does not find the Accept-Language header variable in the user's browser, it looks in the obnls.lst configuration file for the default language of the NetPoint installation.

The obnls.lst file is located in the *COREid_install_dir/identity/oblix/config* directory. *COREid_install_dir* is the directory where COREid is installed.

For more information, see “Managing Multiple Languages” on page 283.

Configuring COREid Server Settings

Configuring a COREid Server includes installing licenses to allow users to use NetPoint, specifying the duration of user sessions, specifying email addresses for user feedback on their experience with NetPoint, configuring mail servers for notification events, managing the cache, and enabling multiple languages.

You use the COREid System Console to view and modify server settings:

- “Configuring Session Timeout” on page 278
- “Configuring Licenses” on page 279

- “Updating License Keys on Multiple COREid Servers” on page 280
- “Customizing Email Destinations” on page 280
- “Configuring a Mail Server” on page 281
- “Managing Caches” on page 283
- “Managing Multiple Languages” on page 283

To view or modify server settings

1. In COREid System Console, click System Admin > System Configuration > View Server Settings.

The View Server Settings page appears, which looks something like the one below.

View Server Settings

This page contains the list of all settings, used by the product. Click any link to change a particular value. You must restart every COF server before the new values can take effect.

[Configure session timeout](#)
180 Minutes

[Configure licenses](#)
Currently installed licenses:

User Manager Vers 7.0.1	Expires on November 16 2004	100 Users
Group Manager Vers 7.0.1	Expires on November 16 2004	100 Users
Organization Manager Vers 7.0.1	Expires on November 16 2004	100 Users

[Customize email destinations](#)

Bug Reports	nalk
Feedback	nalk
Webmaster	nalk

[Mail Server](#)

Server Name	qamail
Server Port Number	25
Domain name	qalab.oblix.com
Mail Send Style	Asynch
Mail Queue Size	100
Mail Style	Supports Rich HTML email.

[Cache](#)

Cache enabled	Yes
----------------------	-----

2. To view or modify a value for a setting, click the link for the setting.
3. Make the changes you want, if any.
4. Click Save to save your changes (or Cancel to exit without saving your changes).
5. Restart the COREid Server for the new values to take effect.

Configuring Session Timeout

Configuring session timeout enables you to specify user-idle session time (in minutes). The user session automatically ends when the specified idle time elapses.

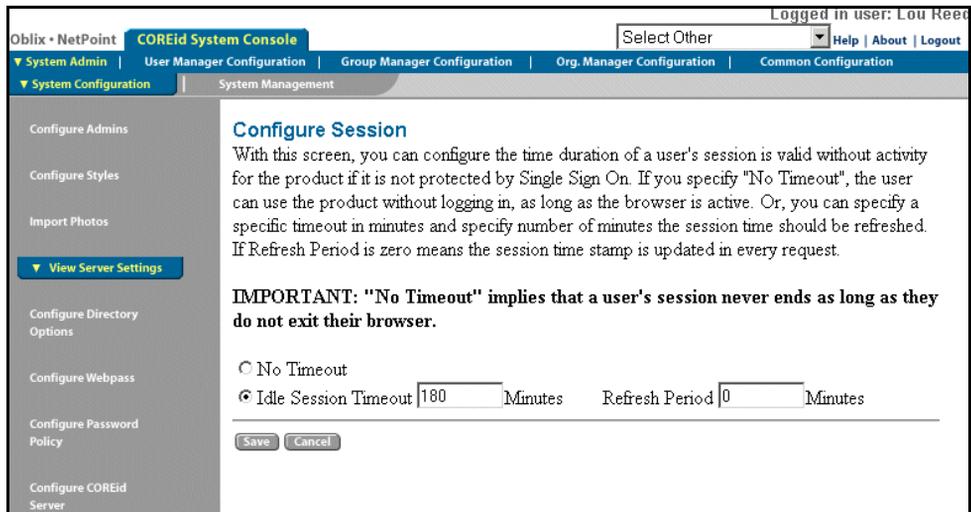
The setting in this page applies to all users and all COREid applications. You cannot have different settings for different users and applications.

A session timeout does not apply if you are using a Web-server-based login, such as through a WebGate, because the WebGate instance handles the timeout.

Note: Resources protected by Web single sign-on always ignore idle session timeout settings.

To configure the length of a user's COREid system session

1. In the View Server Settings page, click Configure session timeout to display this page.



2. Choose a timeout option:

- **No Timeout**—NetPoint sessions continue indefinitely as long as the browser is active.
- **Idle Session Timeout**—The number of minutes Netpoint waits before ending an idle session. After this period of inactivity elapses, the user must log in to the application to continue.

There are several reasons for ending an idle session after a predetermined time period. A short session protects users who leave their workstations without locking them, making them vulnerable to unauthorized use.

- **Refresh Period**—Configures how often NetPoint updates the user session time stamp. A value of 0 (zero) means the session time stamp is updated on every request. Oblix recommends you set this value to 1/4 of the Idle Session Timeout value.

3. Click Save to save your changes (or Cancel to exit the page without saving).

Configuring Licenses

When you install a new NetPoint application, a 60-day license is provided automatically. Oblix provides you with the license keys.

To view or enter a license key

1. From the COREid System Console, click System Admin > System Configuration > View Server Settings.
2. In the View Server Settings page, click Configure licenses to display the following page.

Oblix • NetPoint **COREid System Console** Logged in
 Select Other

▼ System Admin | User Manager Configuration | Group Manager Configuration | Org. Manager Configuration | Common Configuration

▼ System Configuration | System Management

Configure Admins
 Configure Styles
 Import Photos
 ▼ View Server Settings
 Configure Directory Options
 Configure Webpass
 Configure Password Policy
 Configure COREid Server

Configure Product Licenses

To use Oblix NetPoint, you must have a valid License Registration Key. This screen prompts you to enter this Key. Y every COREid server before the new values can take effect.

The License Registration Key should have come with the product. If you do not have a valid license, please contact O 861-6800 or sales@oblix.com to obtain one.

Installed Licenses

User Manager Vers 7.0.1

Group Manager Vers 7.0.1

Organization Manager Vers 7.0.1 _____

COREid Connector for MIIS Vers 7.0.1

Save Cancel

This page displays two types of information:

- **Installed Licenses**—Installed NetPoint applications and their licenses
- **New Licenses**—NetPoint applications that have not yet been licensed

In this sample page, the actual license keys are partially covered.

When you purchase NetPoint, you receive a 60-day temporary license. To obtain a permanent license, contact Oblix Customer Care.

3. After you obtain your permanent license, copy it into the appropriate field.
4. Click Save to save your changes.

Updating License Keys on Multiple COREid Servers

Since the COREid Servers do not talk to each other, updating the license keys in the Administration Console only updates the COREid Server currently handling the request.

To update the license key on multiple COREid Servers

1. After updating a license, identify the license file with the latest time stamp in the following location:

`COREid_install_dir/identity/object/config/license`

where `COREid_install_dir` is the directory where the COREid Server is installed.

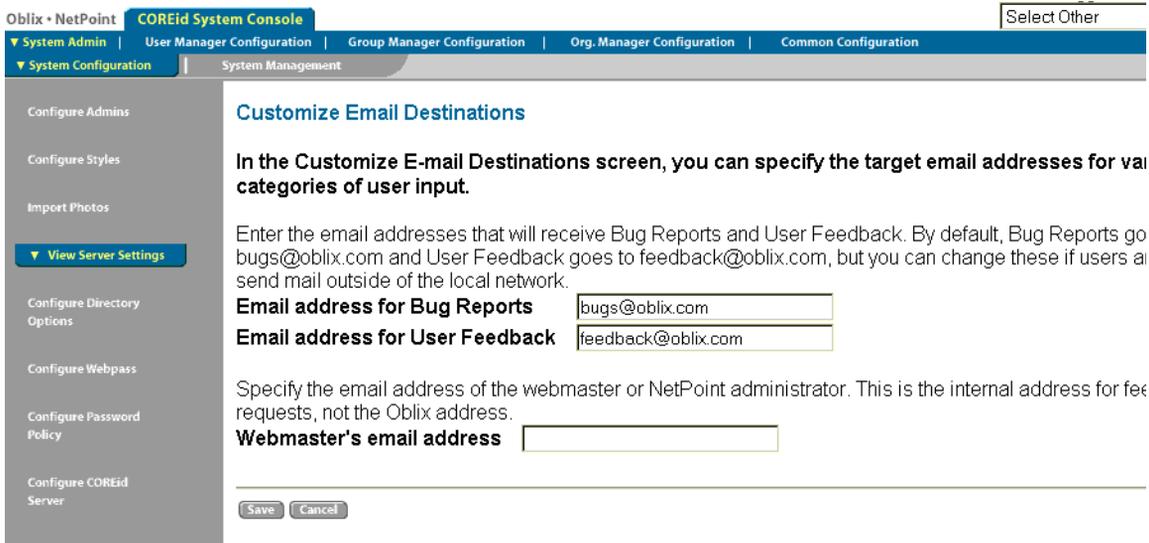
2. Copy that file to the other COREid Servers.
3. For the new licenses to be put into effect, restart the other COREid Servers.

Customizing Email Destinations

Use the Customize Email function to specify email addresses for user feedback. End users access these addresses by clicking About on the side navigation bar, then clicking Submit Admin Feedback or Submit NetPoint Feedback.

To customize email destinations

1. From the COREid System Console, click System Admin > System Configuration > View Server Settings.
2. In the View Server Settings page, click Customize Email Destinations to display this page.



3. Type email addresses for the following fields:
 - **Email address for Bug Reports**—You must change this address if you plan to send it to a person or alias in your organization. This person or department can either solve the problem or contact Oblix for help.
 - **Email address for User Feedback**—If users in your company cannot send email outside the local network, you can type an internal address in the Bugs and Feedback fields. Provide the address of a user who is responsible for forwarding the information to Oblix.
 - **Webmaster’s email address**—Type the email address of the user in your company responsible for administering NetPoint.
4. Click Save to save your changes (or Cancel to exit the page without saving).

Configuring a Mail Server

NetPoint can issue emails alerts during request ticket processing and group management, notification of password expiration, or modification of a Profile attribute. Use the SMTP server configuration function to configure how NetPoint handles these emails.

When configuring a mail server, one of your options is *Supports MHTML email*. MHTML stands for MIME encapsulation of aggregate documents, such as HTML.

MHTML lets you send an HTML document with in-line graphics, applets, and linked documents in a MIME multipart/related body format. You can provide links to other parts included in the HTML document by using the CID (content-ID) URLs or any other kind of URL. The linked body part is identified in its header by either a content-ID (linked to by CID URLs) or a content-location (linked to by any other kind of URL).

The main difference between HTML and MHTML is that with MHTML, graphics are in-line in the email instead of referenced with a link as in HTML format.

To configure a mail server

1. From the COREid System Console, click System Admin > System Configuration > View Server Settings.
2. In the View Server Settings page, click Mail Server to display this page.

SMTP Server configuration

Server Name	<input type="text" value="mail.mailserver.net"/>
Server Port Number	<input type="text" value="25"/>
Domain name	<input type="text"/>
Mail Send Style:	
<input type="radio"/> Synchronous Mailer.	
<input checked="" type="radio"/> Asynchronous Mailer.	Mail Queue Size <input type="text" value="100"/>
Mail Style:	
<input checked="" type="radio"/> Supports Text-only email.	
<input type="radio"/> Supports Rich HTML email.	
<input type="radio"/> Supports MHTML email.	
<hr/>	
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

3. In the Server Name field, enter your SMTP server name.
4. In the Server Port Number field, type the mail server's port number.
5. In the Domain name field, type the Web server's domain name.

Note: This field is optional, but specifying the domain name allows the SMTP connection to be set up according to RFC 821.

6. Select an option under Mail Send Style:
 - **Synchronous Mail**—Sent from the process, such as Modify Attribute, that triggered the email. If an error occurs connecting to the mail server or the server is down, the email is not sent and cannot be regenerated.
 - **Asynchronous Mail**—Uses a thread to queue emails from all applications and sends them one at a time. If the mail server cannot be reached, the thread

re-sends the email. Queued mails are saved to disk. If you select Asynchronous Mailer, specify the mail queue size.

7. Select an option under Mail Style.
8. Click Save to save your changes (or Cancel to exit the page without saving).

Managing Caches

You can view the contents of the COREid Server cache, load the cache with new information, and clear the memory cache to resolve inconsistencies.

To view COREid System cache details

1. From the COREid System Console, click System Admin > System Configuration > View Server Settings.
2. In the View Server Settings page, click Cache to display the page.
3. Select the option you want to view the cache contents, or load or clear the memory cache.

See the *NetPoint 7.0 Deployment Guide* for more information about managing caches.

Managing Multiple Languages

In new installations, the Multi-Language feature is enabled by default. You can enable, disable, and specify preferred languages in the COREid System Console.

Note: When you upgrade from an older version to NetPoint 7.0, the Multi-Language feature is disabled. You must enable this feature from System Configuration > View Server Settings > Manage Multiple Languages page.

To manage a language

1. In the View Server Settings page, click Multi-Language.

The Manage Multiple Languages page appears. Details such as available languages, the order of preference, and whether a language is enabled or not appear on this page.

2. Determine which languages to enable or disable.
 - Select a language and click Enable to enable it.
 - Select a language and click Disable to disable it.
3. Click Back to go back to the View Server Settings page.

See also “Configuring Multiple Languages for NetPoint” on page 272.

Managing COREid Servers

Managing COREid Servers consists of tasks such as adding or deleting COREid Servers, and modifying a COREid Server's parameter values. See the *NetPoint 7.0 Installation Guide* for details about installing a COREid Server. To remove a server completely, you must un-install it.

Topics here include:

- “Setting Up Multiple COREid Servers” on page 284
- “Adding a COREid Server” on page 285
- “Viewing and Modifying COREid Server Parameters” on page 288
- “Deleting COREid Server Parameters” on page 288
- “Managing a COREid Server Service from the Command Line” on page 289

Setting Up Multiple COREid Servers

The following overview outlines how to set up multiple COREid Servers.

Task overview: Set up multiple COREid Servers

1. Install the first COREid Server and a WebPass, then set up the COREid System as described in the *NetPoint 7.0 Installation Guide*.
2. Add a new COREid Server instance in the COREid System Console, as described in the procedure “Adding a COREid Server” on page 285.
3. Associate the new COREid Server instance with a WebPass and specify the priority as Primary, as described in “Managing Associations between COREid Servers and WebPass” on page 318.
4. Modify the WebPass instance to set the maximum connections to the appropriate number to communicate with all primary COREid Servers, as described in “Adding or Modifying a WebPass” on page 312.

You must wait at least one minute before step 5 to ensure that the WebPass configuration file, `webpass.xml`, is updated with the new instance information. Otherwise, the WebPass instance may not receive the new information and cannot connect to the new COREid Server instance.

5. Wait at least one minute before stopping all installed COREid Servers.
6. Install the new COREid Server and indicate that this is *not* the first COREid Server for this directory server, as described in the *NetPoint 7.0 Installation Guide*.

You do not need to update the schema again.

7. Set up the new COREid Server you installed, as explained in the *NetPoint 7.0 Installation Guide*.

Adding a COREid Server

When you want to add a new COREid Server instance to your NetPoint installation, use the procedure below.

To add a COREid Server

1. In the COREid System Console, click System Admin > System Configuration > Configure COREid Servers.

The List all COREid Servers page appears with links to existing COREid Servers.

2. Click the Add button.

The Add a New COREid Server page appears, as shown in part below.

The screenshot shows the 'Add a new COREid Server' configuration page. The page has a blue header with 'blix • NetPoint' and 'COREid System Console'. Below the header is a navigation bar with 'System Admin', 'User Manager Configuration', 'Group Manager Configuration', 'Org. Manager Configuration', and 'Common Configuration'. The main content area is titled 'Add a new COREid Server' and contains the following fields:

Name	<input type="text"/>
Hostname	<input type="text"/>
Port	<input type="text"/>
Debug	<input checked="" type="radio"/> Off <input type="radio"/> On
Debug File Name	<input type="text"/>
Transport Security	<input checked="" type="radio"/> Open <input type="radio"/> Simple <input type="radio"/> Cert
Maximum Session Time (hours)	<input type="text" value="24"/>
Number of Threads	<input type="text" value="20"/>
Audit to Database Flag (auditing on/off)	<input checked="" type="radio"/> Off <input type="radio"/> On
Audit to File Flag (auditing on/off)	<input checked="" type="radio"/> Off <input type="radio"/> On
Audit File Name	<input type="text"/>
Audit File Maximum Size (bytes)	<input type="text" value="100000"/>
Audit File Rotation Interval (seconds)	<input type="text" value="7200"/>
Audit Buffer Maximum Size (bytes)	<input type="text" value="25000"/>
Audit Buffer Flush Interval (seconds)	<input type="text" value="7200"/>
Scope File Name	<input type="text" value="/oblix/logs/scopefile.lst"/>
SNMP State	<input checked="" type="radio"/> Off <input type="radio"/> On
SNMP Agent Registration Port	<input type="text" value="80"/>

3. Fill in the Name through Number of Threads fields as follows:

- **Name**—Type the name of the COREid Server.

- **Hostname**—Type the name of the machine on which the COREid Server is running.
- **Port**—Type the port number of the COREid Server.
- **Debug**—Specify whether you want NetPoint to store debug information on the low-level traffic between the COREid Server and the WebPass.
- **Debug File Name**—Type the name and path of the debug file (the default path is *COREid_install_dir/identity/oblix/logs/debugfile.lst* where *COREid_install_dir* is the directory where COREid is installed).
- **Transport Security**—Select the security method used for communications between the WebPass and the COREid Server:

Open—Used if security is not required. No transport security.

Simple—Provides basic security. Communications are encrypted using TLS v1 (Transport Layer Security, RFC 2246). Communicating elements authenticate one another using a password-based mechanism. All elements that use simple security *must* use the same password throughout the installation. NetPoint provides the certificate that performs the authentication.

Cert—Used if you manage an internal Certificate Authority (CA). Communications are encrypted using TLS v1. In addition, each element, both client and server, must present an X.509 certificate when establishing a connection. The certificate must be provided by a third party such as VeriSign.

Note: You must use the `setup_ois` utility to actually change the mode. See “Changing Transport Security Modes” on page 337 for information.

- **Maximum Session Time (Hours)**—Type the maximum period of time that a connection between the WebPass and COREid Server can remain open. When the time expires, the connection closes and a new one is opened.
 - **Number of Threads**—Type the maximum for number of concurrent requests that the COREid Server is allowed.
4. Complete the Audit information for your environment.

For example:

- **Audit to Database Flag (Auditing On/Off)**—Selecting On directs audit information to a configured database. Off is the default
- **Audit to File Flag (Auditing On/Off)**—Selecting On directs audit information to a file whose name you specify in the next field. Off is the default.

- **Audit File Name**—Type the name of the auditing file where the COREid Server’s auditing information is written. The path specified here is relative to the NetPoint installation directory. The default path is:

COREid_install_dir/identity/oblix/logs/auditfile.lst

where *COREid_install_dir* is the directory where COREid is installed.

Note: For IIS deployments, in order for your audit files to be visible, you must grant write permissions to the IIS user (the system user running the Web server) for the %TEMP% and %TMP% directories and to the audit file destination directory.

- **Audit File Maximum Size (Bytes)**—Type the number of bytes an audit file can contain. When this amount is reached, the audit file is time stamped and saved, and a new file is created.
 - **Audit File Rotation Interval (Seconds)**—Type a number representing the number of seconds that elapse before audit file rotation occurs. Rotation means that the audit file is time stamped and a new file is created. The default is 7200. A setting of 0 means that the audit file never times out, and audit information continues to be added to the file.
 - **Audit Buffer Maximum Size (Bytes)**—Type the number of bytes the audit buffer can hold before it is written to disk. The auditing module maintains a buffer in memory to store the auditing.
 - **Audit Buffer Flush Interval**—Type the interval (in seconds) between the times an audit file’s buffer is flushed to disk.
5. Fill in the Scope File Name field as shown below:
Scope File Name—Type the name of the file that logs bug reports. When a bug report is generated, the information displayed on the page also is logged to a file. This parameter specifies the name of the file for Bug Report or OB_SCOPE messages.

6. Enter details for the SNMP state and agent registration port for your environment.

SNMP State—Selecting On enables SNMP monitoring. Off is the default.

Note: Even if SNMP monitoring is On, to retrieve SNMP statistics you must configure your Network Management Station (NMS) to process the information defined in the NetPoint Management Information Base (MIB). See details on the NetPoint SNMP Agent MIB variables, later in this manual.

SNMP Agent Registration Port—The port that the SNMP agent listens on.

7. Click Save to finish defining your new COREid Server (or Cancel to exit without saving).

Viewing and Modifying COREid Server Parameters

You use the procedure below in the COREid System Console to view or modify parameters.

To view or modify a COREid Server's parameters

1. In the COREid System Console, click System Admin > System Configuration > Configure COREid Server.

A list of existing COREid Servers appears, displaying each server's name, hostname, and port number.

2. Click the name of a COREid Server to view its parameters.

The Details for COREid Server page appears. The server's parameters are listed on this page.

3. Click Modify.

The Modify COREid Server page appears.

4. Modify the parameters as necessary.

See "To add a COREid Server" on page 285 for information about each parameter.

5. Click Save to save your changes (or Cancel to exit without saving).

Deleting COREid Server Parameters

You use the procedure below in the COREid System Console to delete COREid Server parameters.

Note: If you delete a COREid Server from the Console, an attempt to start that server from a command line will fail because the COREid Server's parameters have been deleted from the Console.

To delete a COREid Server's parameters

1. In the COREid System Console, click System Admin > System Configuration > Configure COREid Server.

A list of existing COREid Servers appears, displaying each server's name, hostname, and port number.

2. In the List all COREid Servers page, select the COREid Server you want to delete.

3. Click Delete.

4. When asked to confirm your decision, click OK.

The server's name is removed from the list of servers.

Managing a COREid Server Service from the Command Line

You can use the command line tool *config_ois* to manage tasks related to the COREid Server Service in the Windows Service window.

You can install the COREid Server Service and perform other tasks such as starting or stopping the service with the following commands:

Table 32 Commands for *config_ois*

Command	Operation
<code>[-i <i>install_dir</i>]</code>	Specifies the installation directory for the COREid Server Service.
<code>-v</code>	Specifies the Service name.
<code>[-a <start, stop, query, install, remove>]</code>	Specifies the action to be performed.

The example below uses the query command to obtain information on how the COREid Server Service is configured within the Windows Operating System:

```
C: \COREid_install_dir\identity\oblix\apps\common\bin\
config_ois.exe -q -i c:\COREid_install_dir\identity
-v COREid_ServiceName -a query
```

where *COREid_install_dir* is the directory where COREid is installed and *COREid_ServiceName* is the name of the COREid Server Service.

The query displays the following information:

```
Sample_Srv configuration:
Service Type: 0x110
Start Type: 0x2
Err Control: 0x1
Binary path:
c:\NetPoint\identity\oblix\apps\common\bin\ois_server.exe
Load order group:
Dependencies:
Dependencies: Local System
```

Managing Directory Server Profiles

When installing NetPoint components that communicate with a directory server, you specify the directory server with which the component communicates. Each component communicates with the directory for a specific purpose:

- **COREid Server**—When you install a COREid Server, you designate an LDAP directory server where configuration data is to be stored, and you designate where user data is stored. The user data may be on the same directory server where the configuration data is stored, or it may be a different directory server.
- **Access Manager and Access Server**—When installing an Access Manager or an Access Server, you also designate where user data and configuration data are stored. Additionally, you designate where access policy data is stored.
- **MIIS Integration**—For certain features such as MIIS integration, user access profile reporting, or auditing to a database, you need to create (or have the option to create) an RDBMS profile so that NetPoint can link to external, ODBC 3.0-compatible relational databases. An RDBMS profile is distinct from an LDAP directory server profile.

Note: With NetPoint 7.0, you may have user data stored on one directory server type and configuration and policy data stored together on a different directory server type. For data storage details, see the *NetPoint 7.0 Installation Guide*.

The topics below provide more information:

- “About LDAP Directory Server Profiles” on page 290
- “Creating an LDAP Directory Server Profile” on page 291
- “Viewing an LDAP Directory Server Profile” on page 298
- “Modifying an LDAP Directory Server Profile” on page 298
- “Rerunning NetPoint Setup Manually” on page 299
- “Adding Database Instances to LDAP Directory Server Profiles” on page 301
- “Deleting an LDAP Directory Server Instance” on page 305

About LDAP Directory Server Profiles

For each type of data that NetPoint requires—configuration data, user data, and policy data—an LDAP directory server profile identifies the precise location of the data. The location of policy and configuration data is also stored in .xml files for the COREid Server and in .lst files for the Access Server and Access Manager. A directory server profile contains the connection information for one or more

directory servers that share the same namespace and operational requirements for Read, Write, Search, and so on. The connection information includes a name, a domain or namespace to which it applies, a directory type, and a set of operations. A default directory server profile is created automatically each time you install the COREid Server and specify new directory server connection information.

You can create additional LDAP directory server profiles for load-balancing and failover. You can create directory server profiles that correspond to the partitions of your directory information tree (DIT). Partitioning can potentially increase performance by freeing CPU cycles to perform read and write operations on a specific portion of the DIT. This can be especially beneficial in installations with multiple directory servers and machines.

You can also create LDAP directory server profiles that specify different operations for master and replicated copies of the DIT. For example, you could specify that write operations take place only in the master, and the replica can accept only read operations.

Note: You must always support read, search, modify, create, and delete operations for the directory server profile containing the Oblix tree. You cannot create a read-only or write-only directory server profile for the Oblix tree. If you change settings for the Oblix/policy data directory profile, you need to rerun COREid Server and Access Manager setup and reconfigure the Access Server. For details, see “Rerunning NetPoint Setup Manually” on page 299.

For more information, see:

- “Creating an LDAP Directory Server Profile” on page 291
- “Viewing an LDAP Directory Server Profile” on page 298
- “Modifying an LDAP Directory Server Profile” on page 298

Creating an LDAP Directory Server Profile

Figure 10 shows the Configure Profiles page in the COREid System Console.

The top portion of Configure Profiles page shows details for the directory server that contains user data and NetPoint configuration data. The central portion of the page includes links you can use to configure LDAP Directory Server Profiles. The bottom portion of the page includes links to configure RDBMS profiles. For details about RDBMS profiles, see “Managing RDBMS Profiles” on page 305.

Figure 10 Configure Profiles Page

Configure Profiles

The following contains the Oblix Base and Searchbase settings. Click on the link to change a particular value.

→ **Directory Server**

Machine	lanthanum
Port number	7000
Root DN	cn=Directory Manager
Root password	<Not Displayed>
Search Base	o=company,c=us
Oblix base	o=Oblix,o=company,c=us
Directory Server Security Mode	Open
Disjoint Search Base	

The following table contains the list of all Directory Profiles. Click on any link to change a particular profile. You must stop and restart every COREid server before the new values can take effect.

Configure LDAP Directory Server Profiles

→

Name	Name Space	Primary Servers	Secondary Servers
<input type="checkbox"/> default-np70-COREid1-7010	o=company,c=us	default	
<input type="checkbox"/> AccessManager_setup_user_profile	o=company,c=us	default	
<input type="checkbox"/> AccessServer_default_user_profile	o=company,c=us	default	
<input type="checkbox"/> default-np70-COREid2-7011	o=company,c=us	default	

Configure RDBMS Profiles

→

Name	Primary Servers	Secondary Servers
<input type="checkbox"/> OblixMA	OblixMA	

Clicking the Directory Server link displays the Directory Server Configuration page. If you change the communication mode for the directory server, or hostname or port number, you must also change this information on the Directory Server Configuration page and rerun setup. See “Changing Transport Security Modes” on page 337 for details about this type of change.

The middle portion of the Configure Profiles page is titled “Configure LDAP Directory Server Profiles” followed by a list of links to LDAP directory server profiles for user data, configuration data, and policy data. You can click a profile link to review specifications and supported operations for the profile. You can specify all operations or specific operations, as listed in Table 33.

Table 33 Supported Directory Server Operations

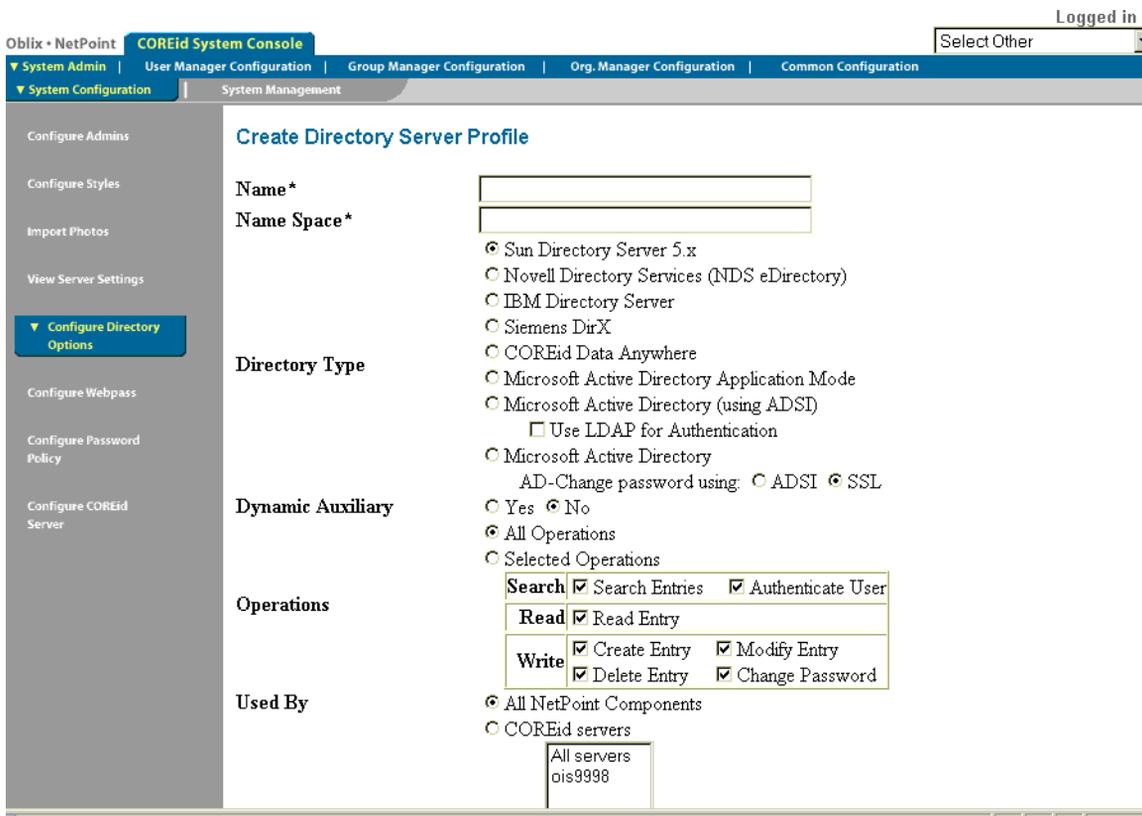
Category	Operation	Comments
	All Operations	All operations are allowed (the default).
Search	Search Entries Authenticate User	The Authenticate User operation allows users to authenticate within the name space of the directory server profile. Selecting this option results in a drop-down list of the login pages for Authentication Domain.
Read	Read Entry	This operation enables the directory server profile to support “Read Schema” as well.
Write	Create Entry Modify Entry Delete Entry Change Password	The Change Password operation allows users to change their password over an ADSI or SSL connection while assigning other more frequently used operations like search to different directory server profiles.

The steps below show how to create a directory server profile.

To create a directory server profile

1. Navigate to COREid System Console > System Admin > System Configuration > Configure Directory Options.
2. Click Add to create a new LDAP directory profile and display the Create Directory Server profile page, shown in part below.

Note: To modify a directory server profile, simply click the name of the profile in the list under Configure LDAP Directory Server Profiles. In this case, the Modify Directory Server Profile Page appears as described in “Modifying an LDAP Directory Server Profile” on page 298.



Note: Fields marked with an * are required.

3. In the Name field, enter a name for the directory server profile.
This name is for informational purposes only. NetPoint uses the naming convention *default-<COREid Server id>* for all default directory server profiles automatically created during COREid installation.
4. In the Name Space field, enter the searchbase for the directory server profile.

Note: Use caution that this namespace does *not* overlap with other directory server profile namespaces. Overlapping namespaces result in duplicate entries. Exceptions to overlapping name spaces include a directory server profile for a Microsoft Active Directory sub-domain, and the directory server profile containing the Oblix configuration DN.

5. Select the type of directory server.

Siemens DirX and Sun—When using either Siemens DirX or Sun (formerly iPlanet) exclusively, you have the option to store data either separately or together, as discussed in the *NetPoint 7.0 Installation Guide*.

COREid Data Anywhere—Requires integrating NetPoint with the OctetString Virtual Directory Engine (VDE).

COREid Data Anywhere is a data management layer that aggregates and consolidates *user data* from multiple sources (including RDBMS and LDAP directories) into a virtual LDAP tree that can be managed by the NetPoint Identity System and used to support authentication and authorization using the NetPoint Access System.

The LDAP directory branches containing NetPoint configuration and policy data must reside on one or more directory servers other than the one hosting VDE or user data. NetPoint applications only recognize configuration and policy information that resides outside the VDE virtual directory.

Important: *Before* installing NetPoint for use with COREid Data Anywhere and VDE, be sure to read the chapter about integrating NetPoint with VDE in the *NetPoint Integration Guide*.

Active Directory—If you select Active Directory, specify whether NetPoint must use ADSI (Active Directory Service Interfaces) for change password operations. Selecting the ADSI option implies you do *not* have to set up an LDAP/SSL connection for password changes. If you do not use ADSI, NetPoint uses an SSL connection to change the password. See “Configuring NetPoint for ADSI” on page 499.

If you have already set up LDAP/SSL for all other regular operations to the directory server, you do not need to set up the certificate server, import the CA certificate, and so forth. Otherwise, you need to configure LDAP/SSL for the password change.

See the *NetPoint 7.0 Installation Guide* for more information.

Dynamic Auxiliary Classes—If you are using dynamic auxiliary classes with Active Directory, select Yes for Dynamic Auxiliary to associate a dynamic auxiliary class with a structural object class in Active Directory 2003.

See “Deploying NetPoint with Active Directory” on page 487 for more information.

Note: You can enable either dynamic or static auxiliary classes in Active Directory 2003.

6. Specify the supported operations for this directory server profile, as listed in Table 33.
7. Indicate which servers are to use this profile.

- **All NetPoint Components**—Select this option if you want each NetPoint component server in this installation to share the same profile.
- **COREid Servers**—Select this option if you want only the COREid Servers to share this profile. If you want a particular COREid Server to use this profile, select the server name from the list box provided.
- **AAA Servers**—The AAA Server option represents the configuration option for the Access Server. You are prompted to create a database profile whenever you add a new Access Server. For details about adding an Access Server instance, see the *NetPoint 7.0 Administration Guide Volume 2*.

The Access Manager directory profile is created during Access Manager setup and is always shared among Access Manager instances.

8. Click Add to associate a directory server instance (database instance) with this profile, and assign the server type as primary or secondary.

See “To add or modify a database instance for an LDAP directory server profile” on page 302 for details.

9. Specify the number of maximum active servers you want (the number of primary and secondary database instances to connect to for load balancing).
 - A default value of 1 indicates that no load balancing takes place.
 - A value greater than 1 distributes database requests across all database instances, depending on which database instance has the shortest job queue. This ensures that the job is processed as quickly as possible.

For more information on load balancing, see the *NetPoint 7.0 Deployment Guide*.

10. Specify the Failover Threshold.

The value specifies the minimum number of primary servers that must be running. If the number of primary servers running goes below the specified number, a failover occurs. It is recommended that this value be the same as the number of maximum active servers. This ensures that failover to any secondary server happens immediately when a primary server goes down.

The default value is 1. This indicates that failover to a secondary server only occurs when there are no primary directory servers to which the COREid Server can connect.

Note: Oblix recommends that this value match the number of maximum active servers to ensure that failover to any secondary server happens immediately when a primary server goes down. For more information on failover and related parameters, see the *NetPoint 7.0 Deployment Guide*.

11. In the Sleep For field, enter the number of seconds before the watcher thread “wakes up” and attempts to re-establish a connection to one or more downed primary servers.

Note: If a primary server is available when failover occurs, the NetPoint COREid Server will fail over to the primary server first.

12. In the Max. Session Time field, specify the number of minutes that the COREid Server keeps a connection to the directory before attempting to reconnect.
13. If this profile is ready for use, select Enable Profile.

14. Save, cancel, or reset as follows:
 - Click Save to save your changes.
 - Click Cancel to exit this page without saving.
 - Click Reset to reset all settings to the default settings.
15. Click OK to confirm your addition.
16. Restart your COREid Servers to enable the new profile.

Viewing an LDAP Directory Server Profile

The middle section of the Configure Profiles page, under the heading Configure LDAP Directory Server Profiles, contains a list of configured directory server profiles.

To view an LDAP directory server profile

1. Navigate to COREid System Console > System Admin > System Configuration.
2. In the System Configuration page, click Configure Directory Options.

The Configure Profiles page appears. The middle section of the page, under the heading Configure LDAP Directory Server Profiles, contains a list of configured directory server profiles.

3. Click the link for the directory server profile that you want to view.

The Modify Directory Server Profile page appears.

Modifying an LDAP Directory Server Profile

There may be occasions when you need to modify an existing LDAP directory server profile.

To modify an LDAP Directory Server Profile

1. Navigate to COREid System Console > System Admin > System Configuration > Configure Directory Options.
2. Click the link for the directory server profile that you want to modify from those listed under the title “Configure LDAP Directory Server Profiles”.
3. Refer to “Creating an LDAP Directory Server Profile” on page 291 for details about parameters.
4. Make the changes you need, then click Save to confirm them.
5. Restart your COREid Servers to enable the new profile.

Rerunning NetPoint Setup Manually

You need to rerun the NetPoint setup after completing any of the following operations on a directory server profile for Oblix and policy data:

- Change directory server configuration options in the System Console.
- Create a new directory profile for Oblix and policy data.
- Delete a directory profile belonging to Oblix and policy data.
- Modify a directory profile for Oblix and policy data.
- Add or change a directory instance within a profile.

Note: You also need to rerun setup when you make specific changes (those marked with an asterisk, *) on the Directory Server Configuration page.

Rerunning setup must occur in a specific sequence.

Task overview: Rerunning system setup

1. Rerun COREid System setup, as described in “Rerunning COREid System Setup” on page 299
2. Rerun Access Manager setup, as described in “Rerunning Access Manager Setup” on page 300, if needed
3. Reconfigure the Access Server, as described in “Reconfiguring the Access Server” on page 301

Rerunning COREid System Setup

Modifying or removing the status parameter in setup.xml tells NetPoint that installation is not complete and permits you to rerun setup.

To rerun COREid System setup

1. Shut down all but one COREid Server if there is more than one running.
2. Go to the only remaining running COREid Server host and open the setup.xml file:

COREid_install_dir/identity/oblix/config/setup.xml

3. Remove the status parameter (or change the status parameter value from “done” to “incomplete”), as shown below:

For example:

```
<NameValuePair ParamName="status" Value="Incomplete"></NameValuePair>
```

4. Save the file.

5. Restart the COREid Server.
6. From your Web browser, launch the COREid System Console.
You will see a Setup page similar to the one that appears during the initial COREid System setup.
7. Initiate setup again and specify the new information.
8. After completing the setup, restart the other COREid Servers.
The other COREid Servers should pick up the new information.
9. Complete the next procedure to rerun Access Manager setup.

Rerunning Access Manager Setup

After rerunning setup for the COREid System, if your implementation includes the Access System, you are ready to setup the Access Manager manually. Modifying or removing the status parameter in `setup.lst` tells NetPoint that installation is not complete and permits you to rerun Access Manager setup.

To rerun Access Manager setup

1. Shut down all but one Access Manager Web server if there is more than one running.
2. Go to the only remaining running Access Manager host and open the `setup.lst` file:

AccessManager_install\dir\oblix\config\setup.lst

3. Remove the status parameter (or change the status parameter value from “done” to “incomplete”), and save the file as shown below:

For example:

```
<NameValuePair ParamName="status" Value="incomplete"></NameValuePair>
```

4. Restart the Access Manager Web server.
5. From your Web browser, launch the Access System Console.
You will see a Setup page similar to the one that appears during the initial Access System setup.
6. Initiate setup again and specify the new information.
7. After completing setup, restart the other Access Manager Web servers.
The other Access Managers should pick up the new information.
8. Rerun Access Server, as described in “Reconfiguring the Access Server” on page 301.

Reconfiguring the Access Server

After manually rerunning setup for the Access Manager, you need to reconfigure the Access Server as indicated below. For additional information on using the `configureAAAServer` tool, see *Volume 2*.

To reconfigure the Access Server

1. Locate the `configureAAAServer` tool.

For example:

```
AccessServer_install_dir/access/oblix/tools/configureAAAServer
```

2. Use the command below with the `configureAAAServer` tool to set up the Access Server.

For example:

```
configureAAAServer install -i AccessServer_install_dir
```

3. Specify new information.
4. Restart your Access Server.

Adding Database Instances to LDAP Directory Server Profiles

A *directory server instance*, which is also known as a *database instance*, contains the bind information for a particular LDAP directory server, including the server name, the host machine, the port, the root DN, and the password. When you define such a database instance, NetPoint validates the configured host and port against the supplied bind credential. Therefore, the directory server corresponding to the database instance must be running when you configure it in NetPoint.

Note: A database instance within an LDAP directory server profile is *not* to be confused with a database instance within an RDBMS profile, which is used to connect NetPoint to an external, ODBC 3.0-compatible relational database. See “Managing RDBMS Profiles” on page 305.

An *LDAP directory server profile* consists of one or more database instances, which are used for load balancing and failover. The directory server profile balances the load among its instances according to the maximum number of active servers; it experiences failover among its instances according to the failover threshold.

Note: Reconfiguring the COREid System to point the configuration directories to a new directory server causes `/CoreID_install_dir/data/common` to be reset. Specifically, in `workflowdbparams.xml`, the parameter `wfinstancenotrequired=true` is reset to `false`. After reconfiguring a directory server instance, manually reset the parameter `wfinstancenotrequired` to `true`.

LDAP Referrals

When you add a directory server instance, you can specify whether or not to enable LDAP referrals. A referral redirects a client request to another server, for the purpose of locating the requested information in another location. A referral contains the names and locations of objects.

If you choose to enable LDAP referrals when you add a directory server instance, you need to set the `enableLDAPReferral` parameter to `true` in the file `ldapconfigdbparams.lst` as shown in the example below for Active Directory:

```
BEGIN: vCompoundList
    specialAttrs:
        BEGIN: vNameList
            userPassword: ( 2.5.4.35 NAME 'userPassword' DESC
' Standard Attribute' SYNTAX ' 1.3.6.1.4.1.1466.115.121.1.5' )
            sAMAccountName: ( 1.2.840.113556.1.4.221 NAME
'sAMAccountName' DESC 'sAMAccountName' SYNTAX
' 1.3.6.1.4.1.1466.115.121.1.15' )
        END: vNameList
        useOIDNamingAttribute: false
        dynamicAuxiliary: false
        enableLDAPReferral: true
    END: vCompoundList
```

To add or modify a database instance for an LDAP directory server profile

1. Navigate to COREid System Console > System Configuration > Configure Directory Options.

The Configure Directory Server Profiles page appears. All the directory server profiles are listed on this page.

2. Click the directory server profile to which you want to add a database instance. The Modify Directory Server Profile page appears.
3. Scroll down to Database Instances and click the Add button (to edit/modify an existing database instance, select it from the list of database instances). The Create Database Instance (or Modify Database Instance) page appears.

Note: The fields for the Modify Database Instance page for an LDAP Directory Server Profile differ from those for the Modify Database Instance page for an RDBMS. For details on RDBMS database instances, see “To add or modify a database instance for an RDBMS profile” on page 309.

Modify Database Instance

Name*	<input type="text" value="default"/>
Machine*	<input type="text" value="lanthanum"/>
Port number*	<input type="text" value="7000"/>
Root DN*	<input type="text" value="cn=Directory Manager"/>
Root password*	<input type="password"/>
Time Limit	<input type="text" value="0"/>
Size Limit	<input type="text" value="0"/>
Flags	<input type="checkbox"/> SSL <input checked="" type="checkbox"/> Referral <input type="checkbox"/> Fast Bind (only for AD on Windows Server 2003)
Secure Port number	<input type="text" value="636"/>
Initial Connections	<input type="text" value="1"/>
Maximum Connections	<input type="text" value="1"/>

Note: The fields marked with an asterisk(*) are required fields.

Changes made to this DB Instance require that you save the DB profile too.

4. Fill in the fields as follows:
 - **Name**— Enter a name for the directory server instance.
 - **Machine**—Enter the name of the machine hosting the directory server instance.
 - **Port Number**—Enter the port number for the directory server.
 - **Root DN**—Enter the Root DN (bind DN) of the directory server user with administrative privileges.
 - **Root Password**—Enter the password of the directory server user with administrative privileges.
 - **Time Limit**—Specify the maximum amount of time allowed for a request to the directory server. The default value is 0 seconds, which means that the server determines the time. The database-instance setting takes precedence over this setting.

- **Size Limit**—Specify the maximum number of entries the directory server can return for a search operation. The default value is 0 entries, which indicates that the server determines the number.
 - **Flags**—Select either of the following:
 - **SSL**—Directory server processes that use SSL. This requires initial certificate configuration. Refer to your directory server documentation for information.
 - **Referral**—Specifies whether the directory server profile should trace LDAP referrals for this directory server. The same bind credentials (Root DN and password) are used to log in to the referral server.
 - **Secure Port**—Specify the port where you access the directory server. Leave this field blank if you are not using SSL or if you are using Active Directory with ADSI for change password.
 - **Initial Connections**—Specify the initial number of connections NetPoint uses to connect to the directory server. These connections are shared among all user requests. The minimum is 1.
 - **Maximum Connections**—Specify the maximum number of connections allowed to the directory server. The default is 1.
5. Do one of the following:
- Click Save to save your settings.
 - Click Cancel to exit without saving your settings.
 - Click Reset to revert to the previous settings.

Deleting an LDAP Directory Server Instance

You may want to remove an LDAP directory server instance.

To delete a directory server instance for an LDAP directory server profile

1. Navigate to COREid System Console > System Configuration.
2. In the System Configuration page, click Configure Directory Options.
The Configure Directory Server Profiles page appears. All the directory server profiles are listed on this page.
3. Click the directory server profile to which you want to add an instance.
The Modify Directory Server Profile page appears.
4. In the Modify Directory Server Profile page, select the Database Instance that you want to delete.
5. Click Delete.

The directory server instance is deleted.

Managing RDBMS Profiles

NetPoint connects to external, ODBC 3.0-compatible relational databases through RDBMS profiles. Currently the MIIS integration, user access profile reporting, and the audit-to-database features make use of such RDBMS profiles. Each profile contains one or more *database instances*, which facilitate database failover if the primary instance of the database goes down.

Note: RDBMS profiles are *not* to be confused with LDAP directory server profiles, which are used to load balance and failover for LDAP directories. The *database instances* within RDBMS profiles are *distinct* from the database instances in LDAP directory server profiles.

The following topics provide additional information:

- “Adding or Modifying an RDBMS Profile” on page 306
- “Adding or Modifying an RDBMS Database Instance” on page 309

Adding or Modifying an RDBMS Profile

The steps to either add or modify an RDBMS profile are similar and are described in the procedure below. The fields you complete are described in Table 34.

Table 34 Field Descriptions for Adding or Modifying an RDBMS Profile

Field	Description
Name	Chose a self-explanatory name for your RDBMS profile.
Used By	Check the box corresponding to the NetPoint feature for which you will be using the RDBMS profile. Currently, the choices are user access privilege reporting, auditing to database, and MIIS integration with NetPoint.
Database Instances	You can create multiple copies of the database for use in failover as follows: <ul style="list-style-type: none">• To add a database instance and click Add. When the Create Database Instance page appears, complete the fields marked by asterisks. For field details, see "To add or modify a database instance for an RDBMS profile" on page 309.• To modify an existing database instance, select it from the database instance list.• To set the server type for the database instance, select Primary or Secondary from the list.• To delete a database instance, check the box next to the instance you want to delete, then click Delete.
Maximum Active Servers	This is the maximum number of servers that can be connected to the relational database at any given time.
Failover threshold	When the number of connected primary servers falls to this number, failover occurs.
Sleep For (Seconds)	Once a connection fails, this many seconds must elapse before failover takes place.
Max. Session Time (Min.)	The connection to the database is discarded after this many minutes, even if it is functioning, and a new connection is established.
Enable Profile	Make sure to check this box if you want the profile to be active.

To add or modify an RDBMS profile

1. Navigate to COREid System Console > System Admin > System Configuration > Configure Directory Options.

The Configure RDBMS Profiles section is at the bottom of the Configure Profiles page.

Configure Profiles

The following contains the Oblix Base and Searchbase settings. Click on the link to change a particular value.

Directory Server

Machine	lanthanum
Port number	7000
Root DN	cn=Directory Manager
Root password	<Not Displayed>
Search Base	o=company,c=us
Oblix base	o=Oblix,o=company,c=us
Directory Server Security Mode	Open
Disjoint Search Base	

The following table contains the list of all Directory Profiles. Click on any link to change a particular profile. You must stop and restart every COREid server before the new values can take effect.

Configure LDAP Directory Server Profiles

Name	Name Space	Primary Servers	Secondary Servers
<input type="checkbox"/> default-np70-COREid1-7010	o=company,c=us	default	
<input type="checkbox"/> AccessManager_setup_user_profile	o=company,c=us	default	
<input type="checkbox"/> AccessServer_default_user_profile	o=company,c=us	default	
<input type="checkbox"/> default-np70-COREid2-7011	o=company,c=us	default	

Configure RDBMS Profiles

Name	Primary Servers	Secondary Servers
<input type="checkbox"/> OblixMA	OblixMA	

2. Select from the RDBMS Profile list the name of the profile you want to edit (or click Add to create a new profile).

Modify RDBMS Profile

Name*

Used By Reporting Auditing MIIS

	Name	Machine	Port number	Server Type
<input type="checkbox"/>	OblixMA		389	Primary <input type="button" value="v"/>

Database Instances

Maximum Active Servers

Failover Threshold

Sleep For (Seconds)

Max. Session Time (Min.)

Enable Profile

Note: The fields marked with an asterisk(*) are required fields.

3. Complete or modify the fields on the Add RDBMS Profile (or Modify RDBMS Profile) page, as described in Table 34.
4. When you are satisfied with the information in the fields, click Save to commit the changes.

Adding or Modifying an RDBMS Database Instance

The steps to create or modify a database instance for an RDBMS database profile are so similar that they are combined in the procedure below. In either case, you must complete fields for the information in Table 35.

Table 35 Field Descriptions to Add or Modify a Database Instance in an RDBMS Profile

Field	Description
Name	The name of the database instance
DSN Name	The name of the DSN for this database instance
Database Name	The name of the database
User name	The name of the administrator with access privileges to this database instance
Password	The password for this database instance
Time Limit	The number of minutes after which the connection to the database is broken and then replaced with a fresh connection
Size Limit	The maximum size of the database
Initial Connections	The number of primary and secondary servers connected to this database instance when it is initialized
Maximum Connections	The total number of primary and secondary Access Servers that can be connected to this database instance

To add or modify a database instance for an RDBMS profile

1. Navigate to COREid System Console > System Admin > System Configuration > Configure Directory Options.

The Configure Profiles page appears.

2. In the Configure RDBMS Profiles section, click Add to create a RDBMS profile (or select from the list the name of the RDBMS profile you want to edit/modify).

Depending on your selection, either the Add RDBMS Profile or Modify RDBMS Profile page appears.

3. In the Database Instances section, click the Add button to create a new instance (or select from the list the name of the instance you want to edit).

4. Complete the fields on the Modify Database Instance or Add Database Instance page.

Field descriptions appear in Table 35.

Modify Database Instance

Name *	<input type="text" value="OblixMA"/>
DSN Name *	<input type="text" value="OblixMA"/>
Database Name	<input type="text" value="OblixMA"/>
User name	<input type="text" value="sa"/>
Password	<input type="password"/>
Time Limit	<input type="text" value="0"/>
Size Limit	<input type="text" value="0"/>
Initial Connections	<input type="text" value="5"/>
Maximum Connections	<input type="text" value="5"/>

Note: The fields marked with an asterisk(*) are required fields.

Changes made to this DB Instance require that you save the DB profile too.

5. Click Save to commit the changes when you are satisfied with the information in the fields on the page.

Configuring WebPass

You first install WebPass after installing the COREid Server. After you set up the COREid System, you can install and configure multiple WebPass instances. Each WebPass instance is installed and configured separately. When a WebPass instance is installed, you supply several required parameters. A NetPoint Administrator can modify these parameters and supply additional information, such as the failover threshold, in the COREid System Console.

A WebPass instance can talk to multiple COREid System servers using the NetPoint Identity Protocol (NIP). You can use one of three transport security modes: Open mode, Simple mode, and Certificate mode, as described in “Changing Transport Security Modes” on page 337.

When a user requests access to a Web server resource, WebPass redirects the request to a COREid Server, which then checks the user's identity through the directory server. You must configure a WebPass plug-in for each Web server.

See the *NetPoint 7.0 Installation Guide* for information about installing WebPass. Topics in this section include:

- “Viewing a Configured WebPass” on page 311
- “Adding or Modifying a WebPass” on page 312
- “Removing a WebPass” on page 315
- “Modifying a WebPass from a Command Line” on page 315
- “Managing Associations between COREid Servers and WebPass” on page 318
- “Disassociating a WebPass from a COREid Server” on page 319

Viewing a Configured WebPass

WebPass configuration occurs using the COREid System Console, Configure WebPass function.

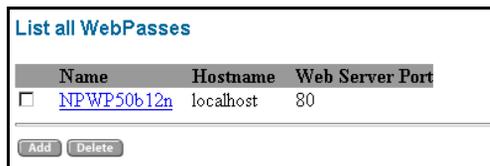
To view a configured WebPass

1. From the COREid System Console, click System Administration > System Configuration > Configure WebPass.

The List all WebPasses page appears. From this page you can add, modify, or delete a WebPass.

2. To view information about a WebPass, click the name of the WebPass.

The Details for WebPass page appears. All the information about the WebPass instance is listed on this page.



	Name	Hostname	Web Server Port
<input type="checkbox"/>	NPWP50b12n	localhost	80

Adding or Modifying a WebPass

Adding a new WebPass involves adding the instance in the COREid System Console, installing WebPass on the Web server host, and updating the Web server configuration to establish communications between the WebPass and the Web server. Use the procedure below to add the instance. See the *NetPoint 7.0 Installation Guide* for other details.

To add a WebPass

1. From the COREid System Console, click System Admin > System Configuration > Configure WebPass.

The List all WebPasses page appears. From this page you can add, modify, or delete a WebPass.

2. From the Configure WebPass page, click Add.

The Add a new WebPass page appears.

The screenshot shows the 'Add a new WebPass' configuration page in the COREid System Console. The page has a blue header with navigation tabs: 'System Admin', 'User Manager Configuration', 'Group Manager Configuration', 'Org. Manager Configuration', and 'Common Configuration'. Below the header is a sidebar with a tree view containing options like 'Configure Admins', 'Configure Styles', 'Import Photos', 'View Server Settings', 'Configure Directory Options', 'Configure Webpass', 'Configure Password Policy', and 'Configure COREid Server'. The main content area is titled 'Add a new WebPass' and contains several input fields: 'Name', 'Hostname', 'Web Server Port', 'Maximum Connections' (set to 1), 'Transport Security' (with radio buttons for Open, Simple, and Cert), 'Maximum Session Time (hours)' (set to 24), 'Failover Threshold', 'COREid Server Timeout Threshold', and 'Sleep For (seconds)' (set to 60). At the bottom of the form are 'Save' and 'Cancel' buttons.

3. In the Name field, type a name for this WebPass instance.

Note: You cannot change the name you save with this instance. To change the name, delete this instance and reconfigure it with a different name.

4. In the Hostname field, type the name of the Web server instance hosting this WebPass.
5. In the Web Server Port field, type the port number the Web server instance is listening to.

6. In the Maximum Connections field, specify the maximum number of connections this WebPass opens to COREid Servers.

The minimum number of connections is 1. You may want to specify more connections for load balancing and failovers.

7. In the Transport Security field, you can modify the security mode that was specified when NetPoint was installed.

The transport security mode specifies the degree of security during communications between the WebPass and the COREid Server. See “Changing Transport Security Modes” on page 337 for more information.

The supported transport security modes are as follows:

- **Open**—No transport security.
- **Simple**—Provides basic security. Communications are encrypted using Transport Layer Security, RFC 2246 (TLS v1). Communicating elements authenticate one another using a password-based mechanism. All elements that use simple security *must* use the same password throughout the installation. NetPoint provides the certificate that performs the authentication.
- **Cert**—Used if you manage an internal certificate authority (CA). Communications are encrypted using TLS v1. Both client and server must present an X.509 certificate from a third party (such as VeriSign) when establishing a connection.

Note: Your COREid Servers and WebPasses *must* use the same transport security mode. Repeat these steps as necessary for *each* installed component.

8. In the Maximum Session Time (hours) field, specify the maximum period of time in hours before the connection between the WebPass and COREid Server is closed and a new one is opened.
9. In the Failover Threshold field, specify the minimum number of connections to Primary COREid Servers.

If this number cannot be met using primary servers, WebPass attempts to do so using secondary servers. For example, if you type 4 in this field, and the number of available connections to primary COREid Servers falls to 3, WebPass attempts to open a connection to a secondary server.

For details about configuring failover between WebPass and the COREid Server, see the *NetPoint 7.0 Deployment Guide*.

10. In the CoreID Server Timeout Threshold field, specify how long (in seconds) the WebPass attempts to contact a non-responsive COREid Server before it considers it unreachable and attempts to contact another.

If a value is not specified, it indicates that there is no timeout.

- 11.** In the Sleep For (seconds) field, specify the interval at which WebPass checks its connection with COREid.

Along with checking for a minimum number of connections, the same check also tries to re-establish primary server connections when secondary connections are currently in use because the failover threshold was not met.
- 12.** Click Save to add the WebPass plug-in (or Cancel to exit this page without saving).

If you click Save, this WebPass plug-in appears on the List all WebPasses page.
- 13.** Associate the WebPass plug-in with one or more COREid Servers, as described in “Managing Associations between COREid Servers and WebPass” on page 318.

Modifying WebPass Details

See “Adding or Modifying a WebPass” on page 312 for more information on the parameters you will modify.

To modify a WebPass

- 1.** From the COREid System Console, click System Admin > System Configuration > Configure WebPass.

The List all WebPasses page appears. From this page you can add, modify, or remove a WebPass.

- 2.** In the List all WebPasses page, click the name of the WebPass that you want to modify.

The Details for WebPass page appears.

- 3.** Click Modify.

The Modify WebPass page appears.

- 4.** Modify the parameters as needed.
- 5.** Click Save to save your changes (or Cancel to exit this page without saving).

Removing a WebPass

Removing a WebPass means that you remove it from the list of configured WebPass instances. To *delete* a WebPass from the Web server instance, you must uninstall it.

To remove a WebPass

1. From the COREid System Console, click System Administration > System Configuration > Configure WebPass.

The List all WebPasses page appears. From this page you can add, modify, or remove a WebPass.

2. In the List all WebPasses page, select the WebPass instance you want to remove.
3. Click Delete.
4. When prompted, click OK to confirm the action.

The WebPass instance is removed from the list of configured WebPasses.

Note: If you remove a WebPass instance in the COREid System Console but do not run the uninstall program, it will be added to the directory server again when you restart the Web server.

Modifying a WebPass from a Command Line

Occasionally you may need to modify the parameters of a WebPass. You modify some parameters, such as Maximum Session Time and Failover Threshold, through the COREid System Console. You can use the command line tool `setup_webpass` to change other parameters, such as the host machine name and transport security mode.

Typically, you use the command-line tool to change the transport security mode. This tool can be used in both Windows and Solaris installations.

To modify a WebPass through the command line

1. Navigate to `WebPass_install_dir\identity\oblix\tools\setupWebPass`.
where `WebPass_install_dir` is the directory where WebPass is installed.

2. From the setupWebPass directory, run the setup_webpass tool.
You can specify parameters using the commands in Table 36.

Table 36 Commands for setup_webpass

Command	Operation
<code>[-i <i>install_dir</i>]</code>	Specifies the installation directory for the WebPass
<code>[-q] [-n <i>WebPass_ID</i>]</code>	Specifies the WebPass ID
<code>[-h <i>COREid_Server_Host_Name</i>]</code>	Specifies the machine name where the COREid Server is installed
<code>[-p <i>COREid_Server_port_#</i>]</code>	Specifies the port number of the machine where the COREid Server is installed
<code>[-s open simple cert]</code>	Specifies the transport security mode
<code>[-P <i>simple cert mode password</i>]</code>	Specifies the password for simple or cert transport security mode
<code>[-c request install]</code>	Specifies a certificate request or installation

To reconfigure transport security mode through the command line

1. To reconfigure a WebPass transport security mode, run the following command from the command line:

```
setup_webpass -i WebPass_install_dir -m
```

2. Select the transport security mode for WebPass:

If you select Open. . .	If you select Simple. . .	If you select Cert. . .
The transport security mode is reconfigured to run in Open mode.	The system prompts you for the password.	<ul style="list-style-type: none"> • The system prompts you for the certificate password. Enter the password at the prompt. • The system prompts you to specify whether you want to request a certificate or install a certificate. • If you specify a certificate request, the system prompts you for the following organization information: Country name State or Province Locality Organization name Organizational unit Common name (for example, HostName.DomainName.com) Email address

- For Cert mode, after you enter the above information, a certificate request is generated and placed in *COREid_install_dir*\identity\oblix\config\ois_req.pem file.

where *COREid_install_dir* is the directory where the NetPoint COREid System is installed.

You must have this certificate request signed by the Certificate Authority.

- If you specify a certificate installation, the system prompts you for the full paths to the location of the Certificate Key file, the Certificate file, and the Certificate Chain file.

After you specify the paths, the transport security mode is reconfigured. For more information, see “Changing Transport Security Modes” on page 337.

To change the transport security mode password

1. Run the following command from the command line:

```
setup_webpass -i WebPass_install_dir -k
```

2. Enter the following information:

- The old password
- The new password
- Reconfirm the new password

The password is changed.

Managing Associations between COREid Servers and WebPass

You must select one or more COREid Servers to receive requests from a WebPass. A single COREid Server can be associated with multiple WebPasses. You can view a list of primary and secondary COREid Servers that are associated with a WebPass instance. You can also modify the number of connections that have been configured between a COREid Server and a WebPass for load balancing and failover purposes, as described in the following procedures:

- “To view COREid Servers associated with a WebPass” on page 318
- “To modify a COREid Server’s connections to a WebPass” on page 318
- “To associate a COREid Server with a WebPass” on page 319

To view COREid Servers associated with a WebPass

1. From the COREid System Console, click System Administration > System Configuration > Configure WebPass.

The List all WebPasses page appears. From this page you can add, modify, or delete a WebPass.

2. In the List all WebPasses page, click the name of a WebPass.

The Details for WebPass page appears.

3. Click List Identity Servers.

The page lists the primary and secondary servers, if any, configured for the WebPass.

4. Click the name of the COREid Server to view details for it.

The Details for COREid Server page appears.

To modify a COREid Server’s connections to a WebPass

1. From the COREid System Console, click System Admin > System Configuration > Configure COREid Server.

The List All COREid Servers page appears.

2. Click a link for the appropriate server.

The Details for COREid Server page appears.

3. In the Details for COREid Server page, click Modify.

The Modify COREid Server page appears, listing the COREid Server details.

4. Change the value in the Number of Connections field, as needed.

5. Click Save to save your changes (or Cancel to exit the page without saving your change).

To associate a COREid Server with a WebPass

1. From the COREid System Console, click System Administration > System Configuration > Configure WebPass.

The List All WebPasses page appears.

2. Click a link for the appropriate WebPass.
3. In the Details for WebPass page, click List COREid Servers.

The next page lists the Primary and Secondary servers associated with the WebPass.

4. Click Add.

The Add a new COREid Server to the WebPass page appears.

5. In the Select Server drop-down list, select a COREid Server.
6. Indicate whether this COREid Server is a Primary or Secondary server.

This information is required for load balancing and failovers.

7. In the Number of connections box, specify the maximum number of connections the WebPass instance opens to this COREid Server.

The minimum is 1. You may want to add more connections for load balancing and failovers.

8. Click Add to associate this COREid Server with the WebPass (or Cancel to exit this page without associating the COREid Server).

Disassociating a WebPass from a COREid Server

Occasionally, you may need to disassociate a WebPass instance from a COREid Server. For example, the machine resources in your division may be reallocated. In this scenario, the associations between WebPass and a COREid Server may no longer be valid. So you must disassociate them from each other. If you do not disassociate them, WebPass continues to poll for the COREid Server and slows down the Web server's performance.

Note: You cannot disassociate a COREid Server if it is the only primary server configured for a WebPass.

To disassociate a COREid Server from a WebPass

1. From the COREid System Console, click Configure WebPass.
2. Click an existing WebPass.
3. Click List COREid Servers.
4. Select the check box next to the COREid Server you want to disassociate.

5. Click Delete.
6. When prompted, click OK to confirm your decision.

The WebPass instance will no longer communicate with the COREid Server.

Configuring Password Policies

A password policy is a set of rules for a legal password such as how long the password is valid, how a user is notified of an upcoming password expiration, how many tries a user is allowed before a password is locked out, and other enforcement parameters. You can create multiple password policies to build a password security system that matches your needs.

You create password policies in the COREid System. The COREid System also includes a lost-password feature that allows an end user to enter a predetermined response to a challenge phrase when the user forgets a password. When he or she enters the correct response, the individual is redirected to a page where the password must be changed before login is allowed.

Password policies can be used in two ways:

- If you have installed only the COREid System, the password policies apply to users trying to log in to the COREid System.
- If you have installed both the COREid and Access Systems, password policies can also apply to users trying to log in to the Access System or trying to access resources protected by the Access System.

Note: If you want your password policies to apply to the Access System, see “Implementing Password Policies in the Access System” on page 328 for details.

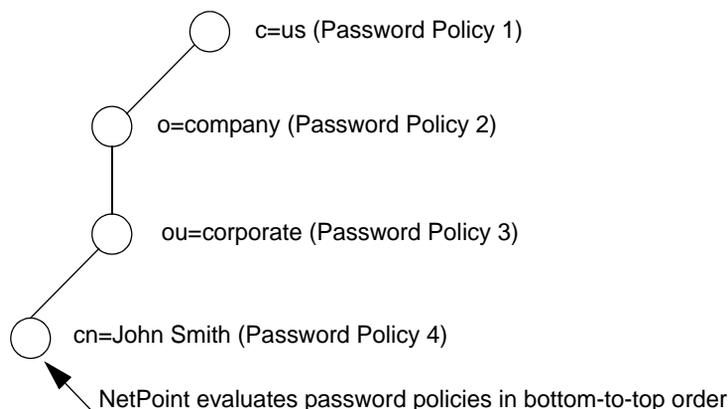
The following sections describe how to create password policies in the COREid System and how to do the additional password management configuration in the Access System:

- “Order of Password Policy Evaluation” on page 321
- “Managing Password Policies” on page 321
- “Configuring Lost-Password Management for the COREid System” on page 327
- “Implementing Password Policies in the Access System” on page 328

Order of Password Policy Evaluation

If a user qualifies under more than one policy within the same domain, the first policy under which the user qualifies is implemented. NetPoint evaluates password policies in bottom-to-top order. NetPoint evaluates and implements the first policy that applies to the user, as illustrated in Figure 11.

Figure 11 Password Policy Evaluation Order



In the above example, all four password policies apply to John Smith, but Password Policy 4 will be implemented because it was evaluated first.

Managing Password Policies

A password policy establishes the basic rules on how to structure and manage passwords. You can create password policies for different directory domains and multiple policies within the same domain. You view, add, modify, and delete password policies in the COREid System Console, as described below:

- “Viewing Password Policies” on page 322
- “Creating Password Policies” on page 322
- “Modifying Password Policies” on page 326
- “Deleting a Password Policy” on page 327

Viewing Password Policies

You can display the Password Policy Management page to access the links to policies you can view.

To view a list of password policies

1. In the COREid System Console page, click System Admin > System Configuration.

2. Click Configure Password Policy.

The Password Policy Management page displays a list of password policies.

3. Click the policy's link to view its settings.
4. Click Cancel to return to the previous page.

Creating Password Policies

During this operation, you can click the Defaults button at the bottom of the page to populate all fields except the Password Policy Name, Password Policy Domain, and Password Policy Filter field with default values.

To create a password policy

1. In the COREid System Console page, click System Admin > System Configuration.

The System Configuration page appears.

2. In the side navigation bar, click Configure Password Policy.

The Password Policy Management page appears.

3. Click Add.

The following page appears.

Oblix • NetPoint **COREid System Console** Logged in user: Lou Reed

Select Other

▼ System Admin | User Manager Configuration | Group Manager Configuration | Org. Manager Configuration | Common Configuration

▼ System Configuration | System Management

- Configure Admins
- Configure Styles
- Import Photos
- View Server Settings
- Configure Directory Options
- Configure Webpass
- ▼ Configure Password Policy
- Configure COREid Server

Password Policy Name

Password Policy Domain

Password policy filter

Password Minimum Length characters

Minimum Number of Uppercase Characters characters

Minimum Number of Lowercase Characters characters

Minimum Number of Nonalphanumeric Characters characters

Externally specified validation rules

Password Validity Period

Password Never Expires

Password Expires in days

Password Expiry Notice Period days before password expires

Mode of Conveying the Expiry Notice

At Login

E-mail

Password minimum age days

Change on Reset

Password History

Do not Keep Password History

Keep Passwords in History

4. In the Password Policy Name field, type the name of your policy.
5. In the Password Policy Domain field, type the domain of the LDAP directory to which this policy applies. For example:
ou=corporate, o=company, c=us
6. In the Password policy filter field, type an LDAP filter to further define the part of the domain to which this password policy applies.

For example, (title=System Administrator) would further limit the availability of this password policy to a subset of people. This field is optional.
7. In the Password Minimum Length field, type the minimum number of characters the password must have.

The default is 8.
8. In the Minimum Number of Uppercase Characters field, type the minimum number of uppercase characters the password must have.

The default is 2.

9. In the Minimum Number of Lowercase Characters field, type the minimum number of lowercase characters the password must have.

The default is 2.

10. In the Minimum Number of Nonalphanumeric Characters field, type the minimum number of nonalphanumeric characters the password must have.

A nonalphanumeric character is any printable character that is not a letter or a number. Examples are +, !, and @.

The default is 1.

11. If external rules apply to this password policy, check Externally specified validation rules.

NetPoint provides an external hook for password policy implementation. Refer to the *NetPoint 7.0 Developer Guide* for information.

12. In the Password Validity Period field, select one of the options:

- Password Never Expires.
- Password Expires In: Enter the number of days this password is valid. The default is 100 days.

13. In the Password Expiry Notice Period, specify the number of days prior to password expiration that users are notified.

14. In the Mode of Conveying the Expiry Notice field, select one or both options:

- **At Login**—When users log in, a message informs them of the number of days remaining until their password expires.

If the COREid System is protected by the Access System, you must enter a Password Expiry Warning Redirect URL. See “Setting Up Password Expiry Warning Redirect URLs” on page 332.

- **E-mail**—Users are notified through email of the number of days remaining before their passwords expire. You cannot customize the message.

15. In the Password minimum age field, enter number of days the password *must* last before users can change it.

16. Select Change on Reset if you want to force users to change the password the first time they log in to the system after an administrator resets the password. By default, the Change on Reset flag is not set.

During self-registration, the Change on Reset flag is not set.

17. In the Password History field, indicate whether or not you want to maintain a password history.

Either select Do not Keep Password History or enter a the number of passwords to be saved for each user. Saved passwords are stored in your LDAP directory and cannot be re-used. The default is 5. NetPoint provides an indexing mechanism that stores passwords in such a way that NetPoint can determine the latest passwords. If you choose to delete one, NetPoint determines which is the oldest.

18. In the Number of Login Tries Allowed field, specify the number of login attempts allowed before NetPoint locks the user's account.

The default value is 3. This means that if a user attempts to login three times using an incorrect login credential, they will be locked out after the third attempt that occurs within the lockout interval specified by "Lockout Duration value". An incorrect login credential consists of a correct username but *incorrect* password. During the lockout interval, the user cannot login even with correct credentials.

Note: This also applies to the number of attempts for a challenge response during Lost-Password Management.

19. In the Lockout Duration field, specify the number of hours an account remains locked after a user exceeds the number of failed logins specified in the previous step.

The default is 24 hours. To clear a lockout before the lockout duration expires, an administrator can reset the user's password from the COREid System. Upon login the user is redirected to a page where he or she can choose a new password—if Change on Reset was selected in the Password Policy Management page before the administrator reset the password.

If Change on Reset was not selected when the administrator assigned a new password, the user can log in to the system with the administrator-assigned password.

20. In the Login Tries Reset field, specify the number of days NetPoint stores the failed login attempts that are uninterrupted by a successful login.

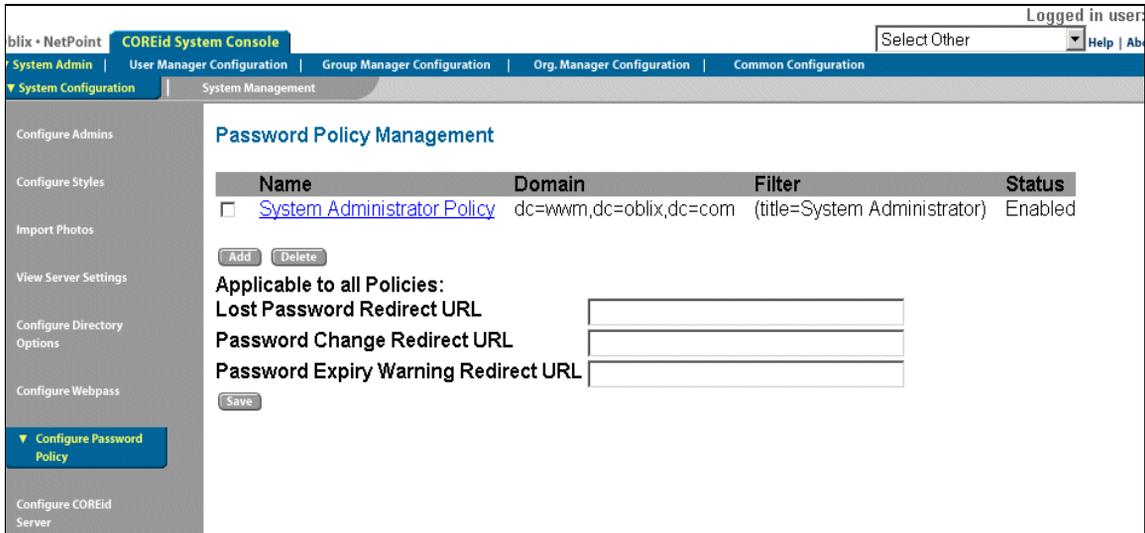
For example, if this value is set to 3, and a user fails to log in once, NetPoint keeps track of that failure for 3 days before clearing it.

21. Select Enable to enable this password policy.

If you later change the setting of this field to enable or disable this password policy, you have to flush the password policy cache. You can flush the cache from the Access System Console > Common Configuration Information > Flush Password Policy Cache. For more information, see *Volume 2*.

22. Click Save to save this policy and return to the Password Policy Management page.

The new policy appears in the list on the page.



Note: The Redirect URLs shown on this page apply to the NetPoint Access System. For more information, see “Implementing Password Policies in the Access System” on page 328.

Modifying Password Policies

During this operation, you can click the Defaults button to populate all fields with default values, except the Password Policy Name, Password Policy Domain, and Password Policy Filter. See “Order of Password Policy Evaluation” on page 321 for information about each parameter.

To modify a password policy's parameters

1. In the COREid System Console page, click System Admin > System Configuration.
2. Click Configure Password Policy.
The Password Policy Management page displays a list of password policies.
3. In the Password Policy Management page, click the policy you want to modify.
The page with the policy's parameters appears.
4. Modify the parameters as necessary.
5. Click Save to save your changes (or Cancel to exit without saving).

Deleting a Password Policy

The Password Policy Management page displays a list of password policies. Saved passwords are stored in your LDAP directory. NetPoint indexes passwords in such a way that NetPoint can determine the latest passwords. If you choose to delete one, NetPoint determines which is the oldest.

To delete a password policy

1. In the COREid System Console page, click System Admin > System Configuration.
2. Click Configure Password Policy.
3. In the Password Policy Management page, select the check box next to the policy you want to delete.
4. Click Delete.
5. Click OK when prompted to confirm your deletion.

The policy is deleted.

Configuring Lost-Password Management for the COREid System

Lost-password management is a link to a Web page that enables users to reset their passwords by responding to a challenge with a correct response. You can enable lost-password management for both the COREid and Access Systems. To support lost-password management, Oblix recommends that your directory administrator extend the schema for your person object class to include two case-insensitive string attributes named Challenge and Response and assign the Challenge and Response semantic types to them. See “Attribute Semantic Types” on page 74 for details.

By default, lost-password management is enabled in the COREid System. You can change this setting using the following procedure to disable the feature.

Also by default, users must enter the Response attribute value through the NetPoint COREid System. This is because the COREid System encrypts the value using a NetPoint encryption scheme licensed from RSA. This encryption scheme is different from secure hash algorithm (SHA). However, you can replace the default encryption with your own by writing a custom action using the Identity Event Plug-in API. This is useful if you already have existing challenge and response attributes that you want to import into the NetPoint COREid System. See the *NetPoint 7.0 Developer Guide* for more information.

Note: You can also implement lost-password management as a portal insert. See the *NetPoint 7.0 Developer Guide* for more information.

Task overview: Implementing Lost-Password Management

1. From the COREid System Console, configure two attributes from your person object class with the following lost-password management semantic types:
 - Challenge
 - Response
2. From User Manager, configure attribute access controls for these attributes, depending on your business rules.

See “Allowing Users to View and Change LDAP Data” on page 126 for more information.
3. From the User Manager, have users or a Delegated Administrator assign values for the Challenge and Response attributes.
4. Users access lost-password functionality by clicking the Lost Password button on the NetPoint COREid System login page.

To enable or disable Lost-Password Management

1. Locate the oblixbaseparams.xml file.

The default path for the file is as follows:

COREidInstall_dir/identity/oblix/apps/common/bin/oblixbaseparams.xml

where *COREidInstall_dir* is the directory where COREid is installed.

2. Make sure Yes is entered for the Apply_LostPwdMgmt parameter.

Yes is the default. Otherwise type No to disable this feature.
3. Save and close the file.

Implementing Password Policies in the Access System

You can apply the password policies configured in the NetPoint COREid System Console to resources that the Access System protects. To do this, you must modify the authentication scheme that protects those resources. Then, when users authenticate to a resource protected by the Access System, the password policy is automatically invoked for those users if they fall within the password policy domain.

Discussions below explain:

- “Modifying Authentication Schemes to Include a Password Policy” on page 329.
- “Configuring Password Redirect URLs” on page 330
- “Entering Password Change Redirect URLs” on page 331
- “Setting Up Password Expiry Warning Redirect URLs” on page 332

- “Updates to the Access Server Cache” on page 332

See the section “Order of Password Policy Evaluation” on page 321 for more information. See the *NetPoint 7.0 Administration Guide Volume 2* for instructions on creating an authentication scheme.

Modifying Authentication Schemes to Include a Password Policy

The following procedure describes how to modify an authentication scheme to include a password policy. You use the Access System Console.

Note: If you make any change to the password policy, be sure to flush the Access Server cache, as described in “Updates to the Access Server Cache” on page 332.

To modify an authentication scheme to include a password policy

1. From the Access System Console, click Access System Configuration and then click Authentication Management in the side navigation bar.

The Authentication Management page appears, listing all the configured authentication schemes.

2. Click the link for an authentication scheme you want to change, and then click the Modify button on the page that appears.

The Modify Authentication Scheme appears.

3. For the `validate_password` plug-in, add the following information to the Plugin Parameters field and then click Save:

```
obReadPasswdMode="LDAP", obWritePasswdMode="LDAP"
```

For example, if the original `validate_password` Plugin Parameters statement for the Basic Over LDAP scheme was:

```
obCredential Password="password"
```

The new parameter set is:

```
obCredential Password="password", obReadPasswdMode="LDAP", obWritePasswdMode="LDAP".
```

The following screen shows the authentication scheme that you would configure:

Details for Authentication Scheme

Name Basic Over LDAP

Description This scheme is Basic over LDAP, using the built-in browser login mechanism

Level 1

Challenge Method Basic

Challenge Parameter realm:LDAP UserName/Password

SSL Required No

Challenge Redirect

Plugin(s)	Order	Plugin Name	Plugin Parameters
	1	credential_mapping	obMappingBase="o=company, c=us",obMappingFilter="(&(objectclass=genSiteOrgPerson)(uid=%userid%))"
	2	validate_password	obCredentialPassword="password",obReadPasswdMode="LDAP",obWritePasswdMode="LDAP"

Update Cache

These parameters must be added for password change redirection.

4. Make a note of the uid parameter value for the credential_mapping plug-in. You need this value when creating the password change redirect URL. For example, the uid parameter value in the previous figure is %userid%.
5. Repeat this process for all other authentication schemes for which you want to set up password change redirection.

Note: If you make any change to the password policy, be sure to flush the Access Server cache. See “Updates to the Access Server Cache” on page 332 for more information.

Configuring Password Redirect URLs

When implementing password policies for the Access System, you can specify a redirect URL for a Web page that notifies users of a password expiration, and a portal insert to the change password functionality.

Note: The lost-password management redirect URL is for informational purposes only. If you are implementing lost-password management for resources protected by the Access System, enter the URL for the corresponding portal insert. This field holds only one value.

Entering Password Change Redirect URLs

You must enter a password change redirect URL if you want users to change their password (either after expiration or after reset). The password change redirect URL is a portal insert for the NetPoint COREid System's change password functionality implemented in a password policy.

To enter a password change redirect URL

1. In the COREid System Console page, click System Admin > System Configuration.
2. Click Configure Password Policy.

The Password Policy Management page displays a list of password policies.

3. In the Password Change Redirect URL, enter a URL with the following syntax:

```
http://machinename:portnumber/identity/obl ix/apps/lost_pwd_mgmt/bin/lost_pwd_mgmt.cgi?program=redirectforchangepwd&l o g i n=%scheme1_uid_parameter_value% %scheme2_uid_parameter_value%%schemeN_uid_parameter_value% &target=top
```

where *machinename:portnumber* are the host and port of the Web server on which a WebPass is installed and *%scheme1_uid_parameter_value% %scheme2_uid_parameter_value%%schemeN_uid_parameter_value%* is the string of uid parameter values for all the authentication schemes for which you want to set up password change redirection.

For example, if you had the following credential_mapping plug-in parameter statements for two different authentication schemes:

- **Form over LDAP**—obMappingBase="o=company, c=us", obMappingFilter="(&(objectclass=genSisteOrgPerson)(uid=%l o g i n%))"
- **Basic over LDAP**—obMappingBase="o=company, c=us", obMappingFilter="(&(objectclass=genSisteOrgPerson)(uid=%useri d%))"

The password change redirect URL would be:

```
http://machinename:portnumber/identity/obl ix/apps/lost_pwd_mgmt/bin/lost_pwd_mgmt.cgi?program=redirectforchangepwd&l o g i n=%l o g i n%useri d%&target=top
```

4. Click Save.

When passwords expire, users are redirected to the specified URL, which displays a NetPoint form for changing a password.

Setting Up Password Expiry Warning Redirect URLs

You must enter a password expiry warning redirect URL if you want to direct users to a custom Web page that warns them that their password is about to expire. Users will be redirected automatically to this URL when the Access Server detects a password expiration warning, but there is no built-in page or portal in NetPoint to serve as a target for this URL.

To set up a redirect URL

1. In the COREid System Console page, click System Admin > System Configuration.
2. Click Configure Password Policy.

The Password Policy Management page displays a list of password policies.

3. In the Password Expiry Warning Redirect URL, enter a URL with the following syntax:

```
http://machinename:portnumber/path-to-custom-page
```

where:

machinename:portnumber are the host and port of the Web server on which a WebPass is installed.

path-to-custom-page is the path of the custom Web page that warns them that their password is about to expire.

4. Click Save.

Updates to the Access Server Cache

You can ensure that the Access Server is notified of changes made by the COREid System and that the Access System's cache is flushed automatically. However, if you choose to not implement automatic cache flush, you can still manually flush the cache when you make changes to the Password Policy Management page in the COREid System. This can be useful in avoiding a significant delay in applying password-policy management changes.

For more information about flushing the Access Server caches, see *Volume 2* and the *NetPoint 7.0 Deployment Guide*

Configuring the Access Server SDK for the COREid System

The Access Server SDK consists of libraries, build instructions, and examples that you use to build an *AccessGate* for non-Web resources. The Access Server SDK is automatically installed with the COREid System in *COREid_install_dir/AccessServerSDK*.

The following functions in the COREid System require the Access Server SDK. You must manually configure the Access Server SDK for these functions:

- Automatic cache flush between the COREid System and Access System
- Automatic login to the Access System after self-registration
- Oblix IDLink

Complete the following procedure if you protect WebPass with a WebGate. You do not have to repeat the procedure for each COREid function previously mentioned.

To configure the Access Server SDK

1. Install and set up the COREid System and Access System, as described in the *NetPoint 7.0 Installation Guide*.

Note: The Access Server SDK is installed automatically with the COREid System in *COREid_install_dir/AccessServerSDK*.

2. From the Access System Console, click Access System Configuration > AccessGate Configuration.
3. Add an AccessGate.

You do not need to configure a port.

The COREid System uses the AccessGate to communicate with the Access Server for purposes of flushing the cache.

For more information about flushing the Access Server caches, see *Volume 2* and the *NetPoint 7.0 Deployment Guide*

4. Select On for Access Management Service.

Access Management Service

Oblix • NetPoint Logged in user: Lou Reed

System Configuration | **NetPoint System Management** | **Access System Configuration**

Add a new NetPoint AccessGate

AccessGate Name	AccessManagerAPI
Hostname	instructor2.oblix.com
Port	
Access Gate Password	
Re-type Access Gate Password	
Debug	<input type="radio"/> Off <input type="radio"/> On
Access Management Service	<input checked="" type="radio"/> Off <input checked="" type="radio"/> On
Maximum user session time (seconds)	3600
Idle Session Time (seconds)	3600
Primary HTTP Cookie Domain	
Preferred HTTP Host	
Maximum Connections	1
Transport Security	<input checked="" type="radio"/> Open <input type="radio"/> Simple <input type="radio"/> Cert
Maximum Client Session Time (hours)	24
Failover threshold	
Access server timeout threshold	
Sleep For (seconds)	60
Maximum elements in cache	100000
Cache timeout (seconds)	1800

5. Save the AccessGate.
6. Access the `COREid_install_dir/identity/AccessServerSDK/oblix/tools/configureAccessGate` directory and run the `configureAccessGate` script. `COREid_install_dir` is the directory where COREid is installed.

See the *NetPoint 7.0 Administration Guide Volume 2* for details about modifying an AccessGate.

When running `configureAccessGate`, ensure that the AccessGate ID is the same as the AccessGate name you entered from the Access System Console in step 3.

7. From the `COREid_install_dir/identity/oblix/data/common` directory, open the `basedbparams.xml` parameter catalog file in a text editor.
8. Change the value of the `doAccessServerFlush` flag to true as shown below:

```
<NameValPair ParamName="doAccessServerFlush" Value="true" />
```
9. Restart the COREid Server.

SECTION III: PERFORMING COMMON ADMINISTRATIVE TASKS

8 Changing Transport Security Modes

Setting up transport security is the subject of this chapter and is one of the administrative tasks that is common to both the COREid System and the Access System.

This chapter contains the following topics:

- “About Transport Security Modes” on page 337
- “Changing Transport Security for the COREid System” on page 342
- “Changing Transport Security Modes for the Access System” on page 349
- “Transport Security Changes for Directory Servers” on page 361
- “Changing Transport Security Passwords” on page 362
- “Importing Multiple CA Certificates” on page 365
- “Changing Access Server Security Password” on page 366
- “Cloned and Synchronized Components” on page 366

About Transport Security Modes

A NetPoint transport security mode is a method to protect communication between two points, such as a client and a server. To ensure protection, communication can be encrypted with a certificate authority (CA).

NetPoint offers the following three transport security modes for communication between NetPoint components, as discussed in greater detail in the *NetPoint 7.0 Installation Guide*:

- **Open**—Communication is not encrypted for protection. Use this mode when security is not an issue; for example, when testing communications between an AccessGate and the Access Server, as long as you consider your network secure. Open is the default setting.

- **Simple**—Communication is encrypted with Netpoint’s internal CA. Simple mode encrypts communications using Transport Layer Security, RFC 2246 (TLS v1). This mode is less secure than Cert mode. Use this mode if you have some security concerns but do not want to manage your own CA.
- **Cert**—Communication is encrypted with an external CA. With Cert mode, communications are encrypted using TLS v1. In addition, each element, both client and server, must present an X.509 certificate (in base64 format) when establishing a connection. The certificate must be provided by you, perhaps from a third-party CA.

Note: With NetPoint 7.0, the default certificate store format and name has changed from cert7.db to cert8.db. When you upgrade to NetPoint 7.0, you continue to use the old certificate store (cert7.db).

When you run the `configureAAAServer`, `setup_ois`, or `setup_accessmanager` utilities, the certificate store format and name is automatically modified to cert8.db. NetPoint 7.0 works with both the cert7.db (upgraded environments) and cert8.db (new installations) certificate store. On non-Windows systems, you use the following tools: `start_configureAAAServer`, `start_setup_ois`, `start_setup_accessmanager`.

NetPoint offers the following two transport security modes for communication between a NetPoint component and the directory server:

- **Open**—Directory server communication is not encrypted for protection. Use this mode when security is not an issue; for example, when testing communications between an AccessGate and the Access Server, as long as you consider your network secure. Open is the default setting.
- **SSL**—Directory server communication using SSL.

Specifying transport security is part of the NetPoint installation process. See the differences when installing the COREid System or Access System, in Table 37.

Table 37 Specifying a Security Mode During Installation

COREid System	Access System
<ul style="list-style-type: none"> • Install the COREid Server component. Specify the transport security mode used to communicate with WebPass. • Install the WebPass component. Specify the transport security mode used to communicate with the COREid Server. <p>See the <i>NetPoint 7.0 Installation Guide</i> for more information on installing NetPoint components.</p>	<ul style="list-style-type: none"> • Install Access Manager. Specify the transport security mode used to communicate with the Access Server. • Create an Access Server instance in the Access System Console. Specify the transport security mode used to communicate with the Access Manager. • Define a WebGate instance in the Access System Console. Specify the transport security mode used to communicate with the Access Server. • Install the Access Server component. Configure the transport security mode to communicate with WebGate. • Install the WebGate component. Configure the transport security mode to communicate with Access Server.

Transport Security Mode Between NetPoint Components

NetPoint transport security can be configured between the following NetPoint components:

- **COREid System**—Transport security between all COREid Servers and WebPass instances must match: either all open, all Simple mode, or all Cert.
- **Access System**—Transport security among all Access Managers, Access Servers, and associated WebGates must match: either all open, all Simple mode, or all Cert.

Access Cache Flushing Caveat—When access cache flushing is enabled on the COREid Server, the COREid Server communicates with the Access Server. In this case, the transport security mode among all five of the following components must be in the same mode.

- COREid Servers and WebPass instances
- Access Managers, Access Servers, and associated WebGates

For details about managing caches, see both “Managing Caches” on page 283 of this manual and *Volume 2*. For more information on caching, see the *NetPoint 7.0 Deployment Guide*.

If you need to change the transport security mode *after* NetPoint is installed, you can change the security mode in the System Console:

COREid System—WebPass and COREid Server—You select a transport security mode for WebPass and COREid Server instances in the COREid System Console. Decide on the type of transport security mode you want to use before you configure WebPass and COREid Server instances. Again, transport security among all COREid components must match. They must all be open, simple, or cert.

Access System—Access Manager, AccessGate, and Access Server—You select a transport security mode for the Access System when configuring AccessGate and Access Server instances in the Access System Console. Decide on the type of transport security mode you want to use before you configure the AccessGate and Access Server instances. Again, transport security among all Access System components must match: either all open, all simple mode, or all cert.

After changing the mode in the System Console, follow the process described in:

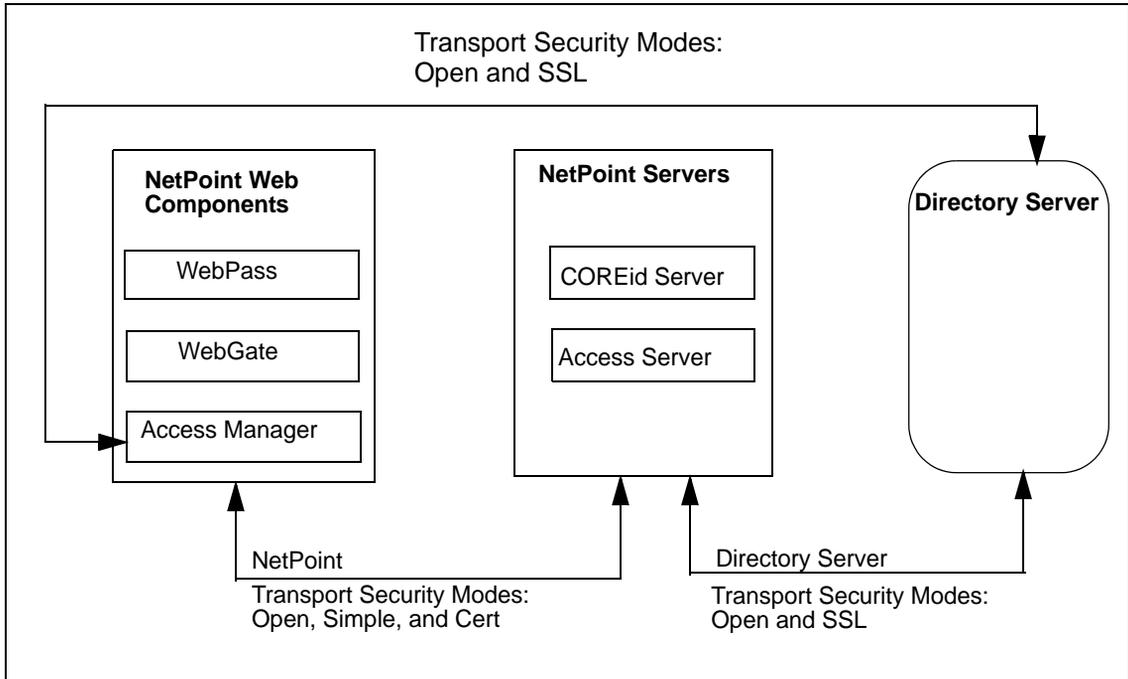
- “Transport Security Changes for Directory Servers” on page 361
- “Transport Security Mode Changes for the Access System” on page 350

You may change the security mode between NetPoint and the Directory server after installation:

COREid or Access Server and the Directory Server—Transport security between the directory server and a COREid or Access Server can be in Open or SSL mode. You specify this transport security mode during installation. If you select SSL, you also specify the location of the SSL certificate. The directory server is automatically updated with the specified security mode information.

The Access Manager is a Web component that reads from and writes to the directory server. You also specify transport security between the Access Manager and directory server. Figure 12 illustrates the supported transport security modes between NetPoint Web components and NetPoint servers, and NetPoint components and the directory server.

Figure 12 Supported Transport Security Modes



You can share directory profiles for all NetPoint components running in SSL mode, even if these components were initially configured in different modes. For example, suppose the COREid Server and Access Server were installed in open mode with the directory, and the Access Manager was installed with SSL enabled for the directory server. In this case, the `cert8.db` and `key3.db` files must exist for each NetPoint component that communicates with the directory server and must reside in the `NetPoint Component_install_dir\identity\access\oblix\config` directory. If these files do not exist, copy them from other existing component folders or run the `genCert` (Access Manager) or other utilities to generate them, as described in this chapter.

About CA Certificates

This discussion explains the root certificate, request, and other certificate files.

If you select the `cert` transport security mode between components during NetPoint installation, you need to create and install a root certificate. The root certificate file is a chain of certificates that is generated when you submit a certificate signing request, such as a CSR to a certificate authority. This request is in the form of an `xxx_req.pem` file. You store a root certificate as a file called `xxx_chain.pem`. You download the `xxx_chain.pem` file from the Certificate Server and store it in the directory below with the `key` and `cert.pem` files, then specify its location during product configuration:

Component_install_dir\identity\access\oblix\config

- Chain file (ois_chain.pem)
- Certificate file (ois_cert.pem)
- Key file (ois_key.pem) the installer may know where this is.

For most NetPoint components, you install certificates during product setup. You install certificates in the Access Manager using the genCert utility. The command for this utility is:

```
genCert -i <install_dir> -m <cert | simple> -P <password> -c <request | install>
```

For example:

```
genCert -i c:\NetPoint\webcomponent\access\oblix\tools\gencert -m cert -P  
<password> -c install
```

You can save an approved certificate to any location that is accessible to the NetPoint component installer. For example, you can save it to /oblix/config.

Note: When using certificates generated by a subordinate CA, the root CA's certificate must be present in the *xxx_chain.pem* along with the subordinate CA certificate. Both certificates must be present to ensure appropriate verification and successful COREid System setup.

The certificate request for WebGate generates the certificate-request file *aaa_req.pem*. You need to send this WebGate certificate request to a root CA that is trusted by the AAA server. The root CA returns the WebGate certificates, which can then be installed either during or after WebGate installation.

The following sections describe cert mode, and requesting and installing certificates.

Changing Transport Security for the COREid System

All COREid Servers and WebPass instances in your installation must run in the same transport security mode. If you specified different modes for different components during your NetPoint installation, you must change them.

Task overview: Changing transport security for the COREid System

1. If you are changing to simple or cert mode, complete certificate preparation.
2. Complete steps in “To change the COREid Server transport security mode” on page 343.

3. Complete steps in “To change the WebPass transport security mode” on page 343.

Note: The WebPass and the COREid Server will not be able to communicate with each other until you have changed the transport security mode for both.

To change the COREid Server transport security mode

1. If you are changing to simple or cert mode, complete certificate preparation.
2. In the COREid System Console, navigate to System Configuration > Configure COREid Server.
3. Select the server you want to modify and click Modify.
4. Click the appropriate button for the transport security mode of your choice. You can select Open, Simple, or Cert mode.
5. Click Save.
6. Restart the COREid Server.

To change the WebPass transport security mode

1. If you are changing to simple or cert mode, complete certificate preparation.
2. In the COREid System Console, navigate to System Configuration > Configure WebPass.
3. Select the WebPass you want to modify and click Modify.
4. Change the transport security mode, and click Save.
5. Stop the WebPass, restart the COREid Server, then restart the WebPass.

Transport Security Mode Changes for the COREid System

When changing the transport security mode after installation, specify the new mode in the COREid System Console, then change the mode in the appropriate configuration files. You repeat the steps shown below as needed for each component.

Table 38 Transport Security Mode Changes for the COREid System

New Security Mode	Process
Open	Specify Open mode in the COREid System Console (see “Changing Transport Security for the COREid System” on page 342).
Simple	<ol style="list-style-type: none">1. Stop the COREid Server.2. Generate the certificate through NetPoint’s internal CA (see “Changing to Simple Transport Security Mode” on page 345).3. Configure the mode in the COREid System Console (see “Changing Transport Security for the COREid System” on page 342).4. Restart the COREid Server.
Cert	<ol style="list-style-type: none">1. Stop the COREid Server.2. Generate the certificate request (see “Changing to Cert Transport Security Mode” on page 346).3. Get the certificate approved through an external CA.4. Install the certificate (see “To install a certificate for Cert mode” on page 347).5. Configure the mode in the COREid System Console (see “Changing Transport Security for the COREid System” on page 342).6. Restart the COREid Server.

Note: The clocks of computers running COREid System components must be synchronized, especially when the components are using open or cert mode. A difference of a few seconds is allowed as long as the COREid Server computer’s clock is ahead of the WebPass computer’s clock. Otherwise, certificate time stamps are invalid, and all requests are rejected. See the *NetPoint 7.0 Administration Guide Volume 2* for details about synchronizing system clocks.

Changing to Simple Transport Security Mode

If you want to change to simple mode, you must first generate a certificate through NetPoint's internal CA.

To generate a certificate through the NetPoint CA

1. Open a Command Prompt window and go to:

COREid_install_dir/identity/oblix/tools/setup

where *COREid_install_dir* is the directory in which the COREid Server is installed.

2. Execute one of the following commands, depending on which component you are modifying:

Operating System	Commands
Unix	<p>COREid Server: start_setup_ois -i <i>COREid_install_dir</i>/identity -m where <i>COREid_install_dir</i> is the directory in which COREid is installed.</p> <p>WebPass: start_setup_webpass -i <i>WebPass_install_dir</i>/identity -m where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.</p>
Windows	<p>COREid Server: setup_ois.exe -i <i>COREid_install_dir</i>\identity -m where <i>COREid_install_dir</i> is the directory in which COREid is installed.</p> <p>WebPass: setup_webpass.exe -i <i>WebPass_install_dir</i>\identity -m where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.</p>

You are prompted to enter simple or cert mode.

3. Type simple and press Enter.
4. Specify and confirm the NetPoint Global Pass Phrase.

This password must be the same across *all* COREid Servers and WebPass instances within a NetPoint installation.

5. Continue with “Changing Transport Security for the COREid System” on page 342.

Changing to Cert Transport Security Mode

If you want to change to cert mode, you must do the following after you install a COREid Server:

- Generate a certificate request to obtain a certificate from an external CA.
- Install the signed certificate after you receive it.

To generate a certificate request for Cert mode

1. Open a Command Prompt window and change to

```
COREid_install_dir/identity/oblix/tools/setup
```

where *COREid_install_dir* is the directory in which COREid has been installed.

For example:

```
cd NetPoint/identity/oblix/tools/setup
```

2. Run one of the commands in Table 39:

Table 39 COREid System Request Certificate Commands

Operating System	Commands
Unix	<p>COREid Server: <code>start_setup_ois -i <i>COREid_install_dir</i>/identity -m</code> where <i>COREid_install_dir</i> is the directory in which COREid is installed.</p> <p>WebPass: <code>start_setup_webpass -i <i>WebPass_install_dir</i>/identity -m</code> where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.</p>
Windows	<p>COREid Server: <code>setup_ois.exe -i <i>COREid_install_dir</i>\identity -m</code> where <i>COREid_install_dir</i> is the directory in which COREid is installed.</p> <p>WebPass: <code>setup_webpass.exe -i <i>WebPass_install_dir</i>\identity -m</code> where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.</p>

You are prompted to enter simple or cert mode.

3. Type cert and press Enter.
4. Indicate that you are requesting a new certificate.
5. Enter information at the prompts for:

- A two-letter country code (the default is US).
- A state or province name.
- Your city or other locality
- An organization name (for example, your company)
- An organizational unit name (for example, your department)
- A common name (for example, your host name)
- An email contact address

6. Press Enter.

You see the message:

“Your certificate request is in the file *COREid_install_dir/identity/oblix/config/ois_req.pem*.”

The *setup_ois* utility creates two files in this directory: *ois_key.pem*, which contains your private key, and *ois_req.pem*.

7. Submit the *ois_req.pem* file to be signed by your Certificate Authority.

To install a certificate for Cert mode

1. Open a Command Prompt window and change to:

COREid_install_dir/identity/oblix/tools/setup

where *COREid_install_dir* is the directory in which COREid is installed.

For example:

`cd NetPoint/identity/oblix/tools/setup`

2. Run one of the commands in Table 40:

Table 40 COREid System Install Certificate Commands

Operating System	Commands
Unix	<p>COREid Server: <code>start_setup_ois -i <i>COREid_install_dir</i>/identity -m</code> where <i>COREid_install_dir</i> is the directory in which COREid is installed</p> <p>WebPass: <code>start_setup_webpass -i <i>WebPass_install_dir</i>/identity -m</code> where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.</p>

Table 40 COREid System Install Certificate Commands

Operating System	Commands
Windows	<p>COREid Server: setup_ois.exe -i <i>COREid_install_dir</i>\identity -m where <i>COREid_install_dir</i> is the directory in which COREid is installed</p> <p>WebPass: setup_webpass.exe -i <i>WebPass_install_dir</i>\identity -m where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.</p>

You are prompted to enter simple or cert mode.

3. Type cert and press Enter.
4. Indicate that you are installing a certificate.
5. Specify the locations of the following files:

ois_key.pem
ois_cert.pem
ois_chain.pem

If you have installed certificates for an earlier NetPoint-generated request, use the default value for ois_key.pem when prompted.

Note: When using certificates generated by a subordinate CA, the root CA's certificate must be present in the ois_chain.pem along with the subordinate CA certificate. Both certificates must be present to ensure appropriate verification and successful COREid System setup.

Your certificate is installed.

6. Continue with “Changing Transport Security for the COREid System” on page 342.

Changing Transport Security Modes for the Access System

Before you change the transport security mode for the AccessGate or Access Server, update the transport security modes for the components in the Access System Console.

You cannot update the transport security mode for Access Manager from the Access System Console. If you are changing from Open mode to another mode, follow the instructions in Table 41 on page 350. If you are changing to Open mode, you need not change the mode for Access Manager because the Access Manager automatically detects that the other AccessGate and Access Server are working in Open mode.

To specify transport security mode for Access Server

1. In the Access System Console, navigate to Access System Configuration > Access Server Configuration.
2. Select the Access Server you want to change, and click Modify.
3. Select the appropriate radio button for transport security, and click Save.
4. Restart the Access Server.

To specify transport security mode for AccessGate

1. In the Access System Console, go to Access System Configuration > AccessGate Configuration.
2. Select the AccessGate you want to change, and click Modify.
3. Select the appropriate radio button for transport security, and click Save.
4. Restart the Web server hosting the AccessGate.

Transport Security Mode Changes for the Access System

You can change the transport security mode for Access System components after you have specified the changes in the Access System Console. The process of changing modes depends on the security mode to which you are changing. If you change an Access Server's security mode, you must change the security mode of all Access Managers and AccessGates pointing to this Access Server to match the new security mode.

If you change the security mode for one or more Access Servers, the Transport Security Mode Change Confirmation page may appear. This page notifies you of an incompatibility between the security modes used by the Access Server and one or more AccessGates.

Note: Configure the Access Server security mode *before* you configure the mode for an AccessGate/WebGate and Access Manager.

Table 41 below lists the process that you follow for each security mode. Repeat these steps as necessary for each installed component.

Table 41 Transport Security Mode Changes for the Access System

New Security Mode	Process
Open	<p>Access Server:</p> <ol style="list-style-type: none">1. Move the appropriate directory or files to a new folder (see "Changing to Open Transport Security Mode" on page 353).2. Configure the Access Server instance in the NetPoint System Console (see "To specify transport security mode for Access Server" on page 349).3. Run the configAAAServer program to specify the new mode. For details about using the ConfigureAAAServer Tool, see the <i>NetPoint 7.0 Administration Guide Volume 2</i>. <p>AccessGate/WebGate:</p> <ol style="list-style-type: none">1. Move the appropriate directory or files to a new folder (see "Changing to Open Transport Security Mode" on page 353).2. Configure the AccessGate instance in the NetPoint System Console (see "To specify transport security mode for AccessGate" on page 349).3. Run the configAccessGate or the configWebGate program, as appropriate, to specify the new mode. To modify an AccessGate through the command line, see the <i>NetPoint 7.0 Administration Guide Volume 2</i>. <p>Access Manager:</p> <ol style="list-style-type: none">1. Restart the Web server on which the Access Manager is installed.

Table 41 Transport Security Mode Changes for the Access System

New Security Mode	Process
Simple	<p>Access Server:</p> <ol style="list-style-type: none"> 1. Move the appropriate directory or files to a new folder (see “Changing to Simple Transport Security Mode” on page 354). 2. Configure the Access Server instance in the NetPoint System Console (see “To specify transport security mode for Access Server” on page 349). 3. Run the configAAAServer program to specify the new mode. For details about using the ConfigureAAAServer Tool, see the <i>NetPoint 7.0 Administration Guide Volume 2</i>. <p>AccessGate/WebGate:</p> <ol style="list-style-type: none"> 1. Move the appropriate directory or files to a new folder (see “Changing to Simple Transport Security Mode” on page 354). You do not need to do this if you are changing over from Open mode. 2. Configure the new mode for the AccessGate instance in the NetPoint System Console (see “To specify transport security mode for AccessGate” on page 349). 3. Run the configAccessGate or the configWebGate program, as appropriate, to specify the new mode. To modify an AccessGate through the command line, see the <i>NetPoint 7.0 Administration Guide Volume 2</i>. <p>Access Manager:</p> <p>Run the genCert utility to specify the new mode. The genCert utility is located in the directory</p> <p><code>AccessManager_install_dir\access\oblix\tools\genCert</code></p> <p>where <code>AccessManager_install_dir</code> is the directory in which the Access Manager is installed.</p>

Table 41 Transport Security Mode Changes for the Access System

New Security Mode	Process
Cert	<p>Access Server:</p> <ol style="list-style-type: none"> 1. Move the appropriate directory or files to a new folder (see “Changing to Cert Transport Security Mode” on page 356)). 2. Configure the Access Server instance in the NetPoint System Console (see “To specify transport security mode for Access Server” on page 349). 3. Run the configAAAServer program to specify the new mode. For details about using the ConfigureAAAServer Tool, see the <i>NetPoint 7.0 Administration Guide Volume 2</i>. <p>AccessGate/WebGate:</p> <ol style="list-style-type: none"> 1. Move the appropriate directory or files to a new folder (see “Changing to Cert Transport Security Mode” on page 356). You do not need to do this if you are changing over from Open mode. 2. Configure the new mode for the AccessGate instance in the NetPoint System Console (see “To specify transport security mode for AccessGate” on page 349). 3. Run the configAccessGate or the configWebGate program, as appropriate, to generate the certificate request and install the certificate. To modify an AccessGate through the command line, see the <i>NetPoint 7.0 Administration Guide Volume 2</i>. <p>Access Manager:</p> <p>Run the genCert utility to specify the new mode. The genCert utility is located in the directory</p> <p><i>AccessManager_install_dir\access\oblix\tools\gencert</i></p> <p>where <i>AccessManager_install_dir</i> is the directory in which the Access Manager is installed.</p>

Changing to Open Transport Security Mode

If you want to change transport security mode from Simple or Cert to Open, run the appropriate configuration program.

To change to Open security mode

1. Move the following directory to a new folder:

AccessSystem_install_dir/access/oblix/config/simple (if in Simple mode)

or

AccessSystem_install_dir/access/oblix/config/.pem* and *password.lst* (if in Cert mode)

where *AccessSystem_install_dir* is the directory in which the Access System components are installed. For example, the Access Manager or Access Server or WebGate.

This saves a previous configuration in case you want to revert to it.

2. Execute one of the commands in Table 42.

Table 42 Access System Commands: Change to Open Mode

Operating System	Commands
Unix	<p>Access Server: <code>start_configureAAAServer reconfig AccessServer_install_dir/access -R</code> where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed.</p> <p>AccessGate: <code>start_configureAccessGate -i AccessGate_install_dir/access -t AccessGate -R</code> where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed.</p> <p>WebGate: <code>start_configureWebGate -i WebGate_install_dir/access -t WebGate -R</code> where <i>WebGate_install_dir</i> is the directory in which WebGate is installed.</p> <p>Access Manager: Run the <code>genCert</code> utility to specify the new mode. The <code>genCert</code> utility is located in the directory <code>AccessManager_install_dir/access/oblix/tools/genCert</code> where <i>AccessManager_install_dir</i> is the directory in which the Access Manager is installed.</p>

Table 42 Access System Commands: Change to Open Mode

Operating System	Commands
Windows	<p>Access Server: configureAAAServer.exe reconfig <i>AccessServer_install_dir</i>\access -R where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed.</p> <p>AccessGate: configureAccessGate.exe -i <i>AccessGate_install_dir</i>\access -t AccessGate -R where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed.</p> <p>WebGate: configureWebGate.exe -i <i>WebGate_install_dir</i>\access -t WebGate -R where <i>WebGate_install_dir</i> is the directory in which WebGate is installed.</p> <p>Access Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>AccessManager_install_dir</i>\access\oblix\tools\genCert where <i>AccessManager_install_dir</i> is the directory in which the Access Manager is installed.</p>

Changing to Simple Transport Security Mode

If you want to implement Simple mode, you do not need to request or install a certificate from an external CA. NetPoint ships with its own internal CA.

To change to Simple security mode

1. Move the following files to a new folder:

AccessSystem_install_dir/access/oblix/config/*.pem

and

AccessSystem_install_dir/access/oblix/config/password.lst (if in Cert mode)

where *AccessSystem_install_dir* is the directory in which the Access System components are installed. For example, the Access Manager or Access Server or WebGate.

This creates a backup file of your older configuration.

2. Generate a certificate through NetPoint's internal CA:

- a) Open a Command Prompt window and change to the appropriate *AccessSystem_install_dir*/access/oblix/tools/*componentDirectory*,

where *componentDirectory* is the directory for the component you are modifying: configureAAAServer, configureWebGate, or genCert (genCert is the utility used by Access Manager).

For example:

cd NetPoint/WebComponent/access/oblix/tools/configureWebGate

b) Execute one of the commands in Table 43:

Table 43 Access System Commands: Change to Simple Mode

Operating System	Commands
Unix	<p>Access Server: start_configureAAAServer reconfig <i>AccessServer_install_dir</i>/access -R where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed.</p> <p>AccessGate: start_configureAccessGate -i <i>AccessGate_install_dir</i>/access -t AccessGate -R where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed.</p> <p>WebGate: start_configureWebGate -i <i>WebGate_install_dir</i>/access -t WebGate -R where <i>WebGate_install_dir</i> is the directory in which WebGate is installed.</p> <p>Access Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>AccessManager_install_dir</i>\access\oblix\tools\genCert where <i>AccessManager_install_dir</i> is the directory in which the Access Manager is installed.</p>
Windows	<p>Access Server: configureAAAServer.exe reconfig <i>AccessServer_install_dir</i>\access -R where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed.</p> <p>AccessGate: configureAccessGate.exe -i <i>AccessGate_install_dir</i>\access -t AccessGate -R where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed.</p> <p>WebGate: configureWebGate.exe -i <i>WebGate_install_dir</i>\access -t WebGate -R where <i>WebGate_install_dir</i> is the directory in which WebGate is installed.</p> <p>Access Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>AccessManager_install_dir</i>\access\oblix\tools\genCert where <i>AccessManager_install_dir</i> is the directory in which the Access Manager is installed.</p>

c) When you are prompted to enter Open, Simple, or Cert mode, select Simple mode and press Enter.

- d) Specify and confirm the NetPoint Global Pass Phrase.

This password must be the same across *all* Access Servers and AccessGates and WebGates. For more information on the Global Pass Phrase, see the *NetPoint 7.0 Installation Guide*.

Important: You need to reinstall the Access Manager if the Simple mode password for the Access Manager is changed, or if the Access System is changed from Simple mode to Cert mode

Changing to Cert Transport Security Mode

The following procedure describes changing the transport security mode to Cert.

Note: The certificate request for WebGate generates the certificate-request file `aaa_req.pem`. You need to send this WebGate certificate request to a root CA that is trusted by the AAA server. The root CA returns the WebGate certificates, which can then be installed either during or after WebGate installation.

To change to Cert security mode

1. Move the following to a new folder:

`AccessSystem_install_dir/access/oblix/config/simple` (if in Simple mode)

This creates a backup of your old configuration.

2. Generate a certificate request.

- a) Open a Command Prompt window and change to the `AccessSystem_install_dir/access/oblix/tools/componentDirectory`

where `AccessSystem_install_dir` is directory in which the Access System components are installed and `componentDirectory` is the directory for the component you are modifying: `configureAAAServer`, `configureWebGate`, `configureAccessGate`, or `genCert` (`genCert` is used by Access Manager)

For example:

```
cd NetPoint/WebComponent/access/oblix/tools/genCert
```

- b) Execute one of the commands in Table 44, depending on which component you are modifying:

Table 44 Access System Request Certificate Commands

Operating System	Commands
Unix	<p>Access Server: start_configureAAAServer reconfig <i>AccessServer_install_dir</i>/access -R where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed.</p> <p>AccessGate: start_configureAccessGate -i <i>AccessGate_install_dir</i>/access -t AccessGate -R where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed.</p> <p>WebGate: start_configureWebGate -i <i>WebGate_install_dir</i>/access -t WebGate -R where <i>WebGate_install_dir</i> is the directory in which WebGate is installed.</p> <p>Access Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>AccessManager_install_dir</i>/access/oblix/tools/gencert where <i>AccessManager_install_dir</i> is the directory in which the Access Manager is installed.</p>
Windows	<p>Access Server: configureAAAServer.exe reconfig <i>AccessServer_install_dir</i>/access -R where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed.</p> <p>AccessGate: configureAccessGate.exe -i <i>AccessGate_install_dir</i>/access -t AccessGate -R where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed.</p> <p>WebGate: configureWebGate.exe -i <i>WebGate_install_dir</i>/access -t WebGate -R where <i>WebGate_install_dir</i> is the directory in which WebGate is installed.</p> <p>Access Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>AccessManager_install_dir</i>/access/oblix/tools/gencert where <i>AccessManager_install_dir</i> is the directory in which the Access Manager is installed.</p>

- c) When you are prompted for a mode, select Cert and press Enter.
- d) Indicate that you are *requesting* a certificate.
- e) Answer the prompts for information, including the following:
- A two-letter country code (the default is US)

- A state or province name
- Your city or other locality
- An organization name (your company, for example)
- An organizational unit name (your department, for example)
- A common name (must be your host machine name)
- An email contact address

f) Press Enter.

A message is displayed stating that your certificate request is in the file *AccessServer_install_dir/access/oblix/config/aaa_req.pem*.

The *setup_aaa* utility actually creates two files in this directory: *aaa_key.pem*, which contains your private key, and *aaa_req.pem*.

g) Submit the *aaa_req.pem* file to the Certificate Authority to get your request signed.

3. Save the approved certificate to a file which the installer can access.
4. Save the CA chain in base64 code format to a *.pem* file that the installer can access.
5. After you receive the certificate from your CA, install the signed certificate.

To install the signed certificate for Cert mode

1. Open a Command Prompt window and change to the *AccessSystem_install_dir/access/oblix/tools/componentDirectory*

where *AccessSystem_install_dir* is the directory in which Access System is installed and *componentDirectory* is the directory for the component you are modifying: *configureAAAServer*, *configureWebGate*, *configureAccessGate*, or *genCert* (*genCert* is the utility used by Access Manager).

For example:

```
cd NetPoint/access/oblix/tools/configureAAAServer
```

2. Execute one of the commands in Table 45:

Table 45 Access System Install Certificate Commands

Operating System	Commands
Unix	<p>Access Server: start_configureAAAServer reconfig <i>AccessServer_install_dir</i>/access -R where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed.</p> <p>AccessGate: start_configureAccessGate -i <i>AccessGate_install_dir</i>/ access -t AccessGate -R where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed.</p> <p>WebGate: start_configureWebGate -i <i>WebGate_install_dir</i>/access -t WebGate -R where <i>WebGate_install_dir</i> is the directory in which WebGate is installed.</p> <p>Access Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>AccessManager_install_dir</i>\access\oblix\tools\gencert where <i>AccessManager_install_dir</i> is the directory in which the Access Manager is installed.</p>

Table 45 Access System Install Certificate Commands

Operating System	Commands
Windows	<p>Access Server: configureAAAServer.exe reconfig <i>AccessServer_install_dir</i>\access -R where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed.</p> <p>AccessGate: configureAccessGate.exe -i <i>AccessGate_install_dir</i>\access -t AccessGate -R where <i>AccessGate_install_dir</i> is the directory in which the AccessGate is installed.</p> <p>WebGate: configureWebGate.exe -i <i>WebGate_install_dir</i>\access -t WebGate -R where <i>WebGate_install_dir</i> is the directory in which WebGate is installed.</p> <p>Access Manager: Run the genCert utility to specify the new mode. The genCert utility is located in the directory <i>AccessManager_install_dir</i>\access\oblix\tools\genCert where <i>AccessManager_install_dir</i> is the directory in which the Access Manager is installed.</p>

3. When you are prompted to enter Simple or Cert mode, type Cert and press Enter.
4. Indicate that you are installing a certificate.
5. Specify the locations of the key, server certificate, and CA chain files:
 aaa_key.pem
 aaa_cert.pem
 aaa_chain.pem

where *aaa* is the name you specify for the file (applicable only to Cert and chain files).

Important: The Webgate certificate request generates the certificate-request file *aaa_req.pem*. You need to send this certificate request to a root CA that is trusted by the AAA server. The root CA returns the WebGate certificates, which can then be installed either during or after WebGate installation.

If you have installed certificates for an earlier NetPoint-generated request, use the default value for *aaa_key.pem* when prompted.

Your certificate is installed.

6. Restart the AccessGate or Access Server, as appropriate.

Transport Security Changes for Directory Servers

When you install the COREid Server and the Access Server, you can specify Open or SSL mode between each of these servers and the directory server. To change the transport security mode after installation, you must reconfigure the COREid Server or the Access Server, as appropriate. During reconfiguration, you can change the security mode between the directory server and the COREid or Access Server.

Note: See the *NetPoint 7.0 Installation Guide* for additional information about adding directory certificates after NetPoint installation.

To change transport security between the COREid Server and directory server

1. From a command line, find the appropriate `setup_ois` tool for your platform.

On Unix, for example:

```
COREid_install_dir/identity/oblix/tools/setup
```

2. At the command prompt, run the appropriate executable.

On Unix, for example:

```
start_setup_ois -i
```

NetPoint takes you through the steps required to set up the COREid Server.

3. When you are asked whether you want SSL between the COREid Server and the directory server, select either y (yes) or n (no).

Note: If you select SSL, provide the full path to the location of the CA certificate when asked.

4. Complete the rest of the steps to finish the reconfiguration process.

To change transport security to SSL between the Access Manager and directory server

1. From a command line, find the appropriate `setup_access_manager` tool for your platform.

On Unix, for example:

```
AccessManager_install_dir/identity/oblix/tools/setup
```

2. At the command prompt, run the appropriate executable to create the cert8.db file.

On Unix, for example:

```
start_setup_access_manager -i
```

NetPoint takes you through the steps required to set up the Access Manager.

3. When you are asked, provide the full path of the file containing the Root CA certificate for the directory server.
4. Complete the rest of the steps to finish the reconfiguration process.

To change transport security between the Access Server and directory server

1. From a command line, navigate to the folder where the configureAAAServer tool is located.

For example:

```
AccessServer_install_dir/access/oblix/tools/configureAAAServer
```

2. At the command line, run the following executable:

```
start_configureAAAServer -i
```

Note: On non-Windows systems, use the start_configureAAAServer tool.

3. Select 1 (Y) to reconfigure the Access Server.

NetPoint takes you through the steps required to set up the Access Server. Specify the required information.

4. When you are asked to specify the mode for the directory server, select either Open or SSL.
5. If you select SSL, provide the full path to the location of the CA certificate.
6. Complete the rest of the steps to finish the reconfiguration process.

Changing Transport Security Passwords

When communicating with each other, NetPoint components authenticate one another using a password-based mechanism.

Simple Mode—In Simple mode, all components in a COREid or Access System must use the same password within the installation. NetPoint generates certificates that are required by Transport Layer Security (TLS). Any NetPoint installation can generate valid certificates.

You can store the password in a local file so that each component can start unattended. Or you may have the component prompt for the password when it starts. Prompting requires a system administrator to start each element manually and type the password.

Cert Mode—Cert mode requires a password for each component’s private key file. You can use a different password for each component.

As with Simple mode, you can store the password in a local file so that each component can start unattended, or you may have the component prompt for the password when it starts. Prompting requires a system administrator to start each component manually and type the password.

You can change the password for Cert or Simple transport security mode.

To change the certificate password for the COREid System

1. Open a Command Prompt window and change to the *COREid_install_dir/identity/oblix/tools/setup* directory, where *COREid_install_dir* is the directory in which COREid is installed.

For example:

```
cd NetPoint/identity/oblix/tools/setup
```

2. Run one of the commands in Table 46:

Table 46 COREid System Commands for Certificate Password Changes

Operating System	Commands
Unix	<p>COREid Server: start_setup_ois -i <i>COREid_install_dir</i>/identity -k where <i>COREid_install_dir</i> is the directory in which the COREid Server is installed.</p> <p>WebPass: start_setup_webpass--i <i>iWebPassInstallDir</i>/identity -k where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.</p>
Windows	<p>COREid Server: setup_ois.exe -i <i>COREid_install_dir</i>\identity -k where <i>COREid_install_dir</i> is the directory in which the COREid Server is installed.</p> <p>WebPass: setup_webpass.exe -i <i>WebPass_install_dir</i>\identity -k where <i>WebPass_install_dir</i> is the directory in which WebPass is installed.</p>

3. Specify the transport security mode this component is using.
4. Specify the old password.
5. Specify and confirm the new password.
6. Restart the COREid Server.

To change the certificate password for the Access System

1. Open a Command Prompt window and change to the *AccessSystem_install_dir/access/oblix/tools/componentDirectory* where *AccessSystem_install_dir* is the directory in which the Access System is installed and *componentDirectory* is the directory for the component you are modifying.

For example:

```
cd NetPoint/access/oblix/tools/configureAccessGate
```

2. Run one of the commands in Table 47:

Table 47 Access System Commands for Certificate Password Changes

Operating System	Commands
Unix	<p>Access Server: start_configureAAAServer chpasswd <i>AccessServer_install_dir/access</i> where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed.</p> <p>AccessGate: start_configureAccessGate -i <i>AccessGate_install_dir/access</i> -t AccessGate -k where <i>AccessGate_install_dir</i> is the directory in which the Access Server is installed.</p> <p>WebGate: start_configureWebGate -i <i>WebGate_install_dir/access</i> -t WebGate -k where <i>WebGate_install_dir</i> is the directory in which the Access Server is installed.</p>

Table 47 Access System Commands for Certificate Password Changes

Operating System	Commands
Windows	<p>Access Server: configureAAAServer.exe chpasswd <i>AccessServer_install_dir</i>\access where <i>AccessServer_install_dir</i> is the directory in which the Access Server is installed.</p> <p>AccessGate: configureAccessGate.exe -i <i>AccessGate_install_dir</i>\access -t AccessGate -k where <i>AccessGate_install_dir</i> is the directory in which the Access Server is installed.</p> <p>WebGate: configureWebGate.exe -i <i>WebGate_install_dir</i>\access -t WebGate -k where <i>WebGate_install_dir</i> is the directory in which the Access Server is installed.</p>

3. Specify the transport security mode this component is using.
4. Specify the old password.
5. Specify and confirm the new password.
6. Restart the Access Server.

Importing Multiple CA Certificates

NetPoint recognizes one CA certificate per directory server type for transport security between a NetPoint component and the directory server for user data, Oblix data, or policy data. If your NetPoint implementation has separate directory servers for user data, Oblix data, or policy data, you can have separate CA certificates for each. Thus you can have up to three CA certificates in your NetPoint implementation; one for the user directory, one for the Oblix directory, and one for the policy directory.

Important: If your installation uses replicated and/or multiple directories that have established SSL using certificates from different certificate authorities, you need to import the various certificates manually into the NetPoint cert8.db file. The cert8.db file is encrypted and stored in a proprietary Mozilla format.

For more information about adding directory server CA certificates, see the *NetPoint 7.0 Installation Guide*.

Changing Access Server Security Password

You can change the Access Server transport security mode from the command line. For Simple mode, the AccessGate or WebGate and the Access Server must have the same password to allow them to communicate with each other.

To change the transport security mode password

1. Run the following executable:

```
configureAAAServer chpasswd AccessServer_install_dir
```

where *AccessServer_install_dir* is the directory in which the Access Server is installed.

2. Specify the following when prompted:

- The transport security mode in which the Access Server is configured.
- The old password
- The new password

3. Restart the Access Server.

See “About Transport Security Modes” on page 337 for more information.

Cloned and Synchronized Components

Instead of using the command line or the installation GUI to install a NetPoint component, you can automatically install a component by *cloning* the configuration of an already-installed component. Cloning creates a copy of a component on a remote system using an already-installed component as a template.

Synchronizing allows you to harmonize two installations of the same NetPoint component when one is more up-to-date than the other. Synchronization can be used to upgrade or repair installations on similar platforms.

See the *NetPoint 7.0 Installation Guide* for more information on cloning and synchronizing.

9 NetPoint Reporting

This chapter provides an overview of NetPoint reporting features, the information each feature presents, the types of output available, and possible uses for these reports. This chapter is organized as follows:

- “About NetPoint Reporting” on page 367
- “Summary of NetPoint Reporting Features” on page 371

About NetPoint Reporting

NetPoint can collect and present a wide range of information related to the following:

- Users and resources in your NetPoint directory
- Activities on the Access and COREid systems
- The operation, administration, and maintenance of your NetPoint system

To help distinguish among the many report-related features built into NetPoint, this chapter reserves certain terms to describe specific functional areas, as explained in the following table:

Table 48 Reserved Terms Used for NetPoint Reporting Features

Feature	Description
Monitoring	Refers exclusively to the SNMP data collected so that you can monitor the health and performance of the network components that host your NetPoint system. For a complete discussion of SNMP Monitoring, see “SNMP Monitoring” on page 455.
Logging	Refers exclusively to program execution data collected so that you can diagnose the health of the components that make up your NetPoint system, troubleshoot execution errors, and debug custom AccessGates and other NetPoint plug-ins. For a complete discussion of NetPoint logging, see “Logging” on page 373.

Table 48 Reserved Terms Used for NetPoint Reporting Features

Feature	Description
Auditing	<p>Refers to two types of data:</p> <ul style="list-style-type: none">• Dynamic audit data is collected from Access Servers and COREid Servers. It encompasses NetPoint system events such as resource requests, password changes, and account revocation.• Static audit data is collected from the NetPoint directory server. It encompasses policy and profile information. <p>For a general discussion of static and dynamic reports, see “Report Types” on page 368.</p> <p>For a complete discussion of NetPoint auditing, see “Auditing” on page 395.</p>
Diagnostics	<p>Refers to parameter settings and state information on Access Servers, COREid Servers, and their connections to the NetPoint directory components. For more on Access System and COREid System diagnostics see “Static Audit Report Types” on page 397.</p>
Access Testing	<p>Refers exclusively to the on-screen display that provides a quick way of determining whether a given user has access to a given resource at a given time. For more on access testing, see “Static Audit Report Types” on page 397.</p>
Filtered Queries	<p>Refers to the advanced searches of the NetPoint directory conducted through various NetPoint applications to generate lists of users or resources that share certain combinations of profile or policy attributes. For more on advanced filtered queries, see: “Static Audit Report Types” on page 397.</p>
NetPoint Audit Reports	<p>Refers exclusively to data that is collected from the NetPoint servers and directory server, stored in the NetPoint audit database, then extracted, compiled, and formatted by preconfigured Crystal Reports presentation templates. For a complete discussion of NetPoint Audit Reports, see “About NetPoint Audit Reports” on page 411 and “Setting up NetPoint Audit Reports” on page 451.</p>

Report Types

The information collected and reported by the various NetPoint reporting features falls into two broad categories:

- **Static reports**—Generally compiled from settings stored on NetPoint components or third-party related components. For example, policy and profile information stored on the NetPoint directory server is classified as static audit data. Connection settings (and states) fall into the Diagnostic category. Certain NetPoint Audit Reports use static (stored) policy and profile information to compile a list of resources that are available to specified users during specified times.

- **Dynamic Reports**—Focus on events and changes in state at various levels throughout the NetPoint system. For example, the logging feature can record each function call (and outcome) originating from a given NetPoint component. This low-level trace capability can be useful to developers. At the other end of the spectrum, the dynamic audit feature can reveal system intrusion threats to NetPoint security administrators by reporting patterns of failed authentication attempts on specific servers during a specific interval.

Data Sources

The NetPoint reporting features can gather data from a variety of sources, the most important of which are covered in Table 49.

Table 49 Primary Data Sources for the NetPoint Reporting Features

Data Source	Description
NetPoint directory	Stores several types of static information, including the following: <ul style="list-style-type: none"> • User, group, and organization profile settings • Policy settings for protecting resources • Connections settings such as those used to connect with NetPoint components or the various databases used by NetPoint • Certain NetPoint security settings • NetPoint schema used to organize the LDAP directory at the heart of the NetPoint system
Component configuration files	Many key settings reside in configuration files stored within the directory structure of the NetPoint component they affect. This can range from the path to a database driver to the size of the buffer used for queuing log output.
System configuration files	These settings for the machines that host the various NetPoint components can be environment variables that make components “visible” to each other, or they can be protocol settings that enable components to communicate at the same level. Generally, NetPoint does not report such system-level configurations directly, but it can sometimes report corresponding NetPoint settings that must match the settings established at the host system level.
Access Servers	In addition to providing configuration information about the settings they maintain to interact with other components, Access Servers can report Access System events such as authorization requests and their outcomes. This information is useful for determining “who has gained (or tried to gain) access to what during a certain interval.”
COREid Servers	COREid Servers also store certain settings that govern how they interact with other components. Additionally, they report COREid System events such as who attempted to submit credentials at what time, and whether that authentication attempt succeeded.

Table 49 Primary Data Sources for the NetPoint Reporting Features

Data Source	Description
Other NetPoint components	Components such as the Access Manager can report changes to policies and certain other NetPoint activities and settings.

Data Output

Generally, the various types of NetPoint reports can send data to one or more of the following three destinations:

- The NetPoint graphical user interface
- A plain text file on the machine hosting the NetPoint component that is sending the data
- A system file on the machine hosting the component that is sending the data
- A central database

Note: When data is sent to the audit database, it is generally filtered, compiled, and presented using special Crystal Reports templates that generate NetPoint Audit Reports.

When a report is sent to the NetPoint graphical user interface, it is likely to be somewhat less extensive than the equivalent type sent to a file or database. For instance, the on-screen Access Tester tool cannot report on the kind of complex user and resource groups that are available through the User Access Privilege tool, which sends output to a plain-text file or the audit database.

Output Configuration

Generally, you can format report output in one or both of the following ways:

- Through the NetPoint graphical user interface
- By manually editing a plain-text configuration file.

In a limited number of cases and to a limited extent, you can configure report output through a third-party GUI. For example, you can edit the templates used to generate the NetPoint Audit Reports through the Crystal Reports interface.

Data Uses

NetPoint reports can prove useful to a wide variety of personnel, including some of the following:

- NetPoint Administrators
- Network administrators
- Security administrators
- Compliance administrators
- Custom AccessGate and NetPoint plug-in developers

Summary of NetPoint Reporting Features

Table 50 provides an overview the NetPoint reporting features, the information they present, and potential uses to which these features can be applied.

Table 50 Overview of NetPoint Reporting Features

Feature	Type	Output	Source	Data	Potential uses
Monitoring	Dynamic	File	SNMP instrumentation	Network component states and events	Monitoring and troubleshooting the network hosting your NetPoint system
Logging	Dynamic	File	NetPoint components	Program execution (states and events)	Diagnosing component health and debugging custom AccessGate and NetPoint plug-in code
Auditing	Dynamic	File, DB	NetPoint servers	System events	Tracking NetPoint usage patterns, NetPoint system performance, component loading, and security compliance
Auditing	Static	File, DB	directory server	Profile and policy attributes	Identifying users and resources that fit specified patterns
Diagnostics	Static	GUI	directory server	Directory component, NetPoint server, and connection settings and states	Verifying NetPoint server and directory server settings, states, and connection details
Access Tests	Static	GUI	directory server	Profile and policy attributes	Quick determination of "who has access to what at a given time."
Filtered Queries	Static	GUI, file	directory server	Profile and policy attributes	Reporting on complex combinations of shared profile and policy attributes

Table 50 Overview of NetPoint Reporting Features

Feature	Type	Output	Source	Data	Potential uses
NetPoint Audit Reports (from Crystal Report templates by way of the audit database)					
Global Access	Static	GUI, file, hardcopy	directory server via audit db	Profile and policy attributes	Advanced reports on user and resource access privileges
Authentication	Dynamic	GUI, file, hardcopy	NP servers via audit db	Authentication events	Statistics on authentication events
Authorization	Dynamic	GUI, file, hardcopy	NP servers via audit db	Authorization events	Statistics on authorization events
Activity	Dynamic	GUI, file, hardcopy	NP servers via audit db	Access and COREid system events	Statistics on and lists of various NetPoint events
ID history	Dynamic	GUI, file, hardcopy	NP servers via audit db	Profile attributes and changes to attributes	Statistics on and lists of identity profile changes

10 Logging

This chapter focuses on NetPoint logging, which is explained by the following topics:

- “About Logging and Log Levels” on page 373
- “About Log Configuration Files” on page 375
- “About Log Writers” on page 380
- “Log Configuration File Structure” on page 381
- “Controlling Logging Levels” on page 384
- “Log Configuration Parameters” on page 386
- “Configuring Logs in the COREid System Console” on page 390

About Logging and Log Levels

The NetPoint logging feature enables you to collect a wide range of NetPoint program execution data so that you can troubleshoot system performance issues and diagnose component health problems.

Logging stands as just one of several features for collecting and presenting NetPoint-related information. For an overview of other reporting features, including NetPoint system event auditing, COREid and Access System diagnostics, and SNMP monitoring, see “NetPoint Reporting” on page 367.

You can control logging activity for NetPoint components by specifying log output for individual Access Servers, COREid Servers, Access Managers, WebPasses, WebGates, custom AccessGates, and custom NetPoint plug-ins.

The parameters that control logging activity reside in configuration files stored with each component. You customize log output for each NetPoint component by manually editing the associated configuration file. For COREid Servers only, you have the option to set certain log parameters through the COREid System Console.

You can send the log data generated by a specific component to either of the following destinations, or neither, or both:

- A log file stored in the directory tree under the root installation directory of the component generating the data.
- The system file of the machine hosting the component logging data. (When more than one NetPoint component resides on the same host, all components can send data to the system log file on that machine.)

For convenience, the many thousands of program events and states reportable through logging are classified within an eight-level, pyramidal hierarchy. At the highest level, the Fatal category includes about 60 catastrophic events that usually force a NetPoint component to exit. At the bottom of the pyramid, the Trace level reports about 900 Oblix API and 150 third-party API calls and their outcomes. In most cases, these Trace level messages are meaningful only to developers and plug-in programmers.

Log Levels

The logging feature can collect logging data at one or more levels of detail. Since each level is activated individually, you can collect data from non-adjacent levels.

The following table lists the eight hierarchical levels that the dominant LOG_THRESHOLD_LEVEL parameter uses to establish which levels can be activated for logging. (For details, see Table 55 on page 384.)

The ninth entry in this table, LOGLEVEL_ALL, encompasses all eight levels in the hierarchy.

Table 51 NetPoint Logging Levels

Level	Number of Events Reported	Description
LOGLEVEL_FATAL	> 60	Critical errors are reported at this level. Generally, these events are serious enough to cause the component to exit.
LOGLEVEL_ERROR	> 960	Events that may require corrective action are written to the log file. For example, an error-level entry is generated when the component is unavailable. An error log entry may also be generated for transient or self-correcting problems, such as failure to connect to another component.
LOGLEVEL_WARNING	> 1200	Issues that may lead to an error or require corrective action at some point in the future are written to the log file.

Table 51 NetPoint Logging Levels

Level	Number of Events Reported	Description
LOGLEVEL_INFO	> 400	Successfully completed actions or the current state of a component (if the component is initializing, for instance) are written to the log file.
LOGLEVEL_DEBUG1	> 400	Basic debugging information is written to the log file. Typically, the information at this log level is only meaningful to a NetPoint developer.
LOGLEVEL_DEBUG2	> 100	Advanced (or rarely needed) debugging information is written to the log file. This log level augments the information provided in the Debug1 log level. Typically, the information at this log level is only meaningful to a NetPoint developer.
LOGLEVEL_DEBUG3	> 900	A large amount of debugging information (or data pertaining to an “expensive” section of the code) is written to the log file. This level is useful for debugging a tight loop or a performance-sensitive function. Typically, the information at this log level is only meaningful to a NetPoint developer.
LOGLEVEL_TRACE	> 900 Oblix API; > 150 3rd-party API	This log level is used to trace code path execution or to capture performance metrics. This information is captured at the entry and exit points for each component function. Typically, the information at this log level is only meaningful to a NetPoint developer.
LOGLEVEL_ALL	> 5000	This amalgamated “level” includes all the events and states from the eight hierarchical levels listed above. Note: Even if you specify LOGLEVEL_ALL, logging may still not be activated at all levels, because the LOG_THRESHOLD_LEVEL takes precedence. See Figure 13 on page 390 for details.

About Log Configuration Files

The parameters that control log output reside in XML-based log files, which you can edit with any plain-text editor.

Log Configuration File Paths

When you install a NetPoint component, a default log configuration file is placed in the following location:

Component_install_dir\identity\access\oblix\config

where *Component_install_dir* is the directory where you are installing the component.

When you install more than one instance of a given component (such as multiple COREid Servers, for example), a logging configuration file is installed for each instance.

Important: To ensure that NetPoint components can find the log configuration file, do not change the default path.

Be aware that a log configuration file is distinct from a log data file. For details on log data files, see Table 57, “Log Data File Configuration Parameters,” on page 387.

Log Configuration File Names

The following table lists the names of the log configuration files for each type of component. To ensure that NetPoint components can find this file, do not change the default name.

Table 52 Log Configuration File Names for NetPoint Components

NetPoint Component	Logging Configuration File Name
Access Server	oblog_config.xml
COREid Server	oblog_config.xml
Access Manager	oblog_config_am.xml
WebGate	oblog_config_wg.xml
WebPass	oblog_config_wp.xml
Access Server SDK (custom AccessGate)	oblog_config.xml

Modifying a Log Configuration File

The parameters set in the log configuration file associated with a given component determine the type of information that gets logged for that component, the destination to which the data gets sent, and in certain cases, the size of the write buffer used for the log and the manner in which the target log file is rotated, among other specifics.

For all components, you edit the XML statements in the log configuration file with a plain text editor. For COREid Servers only, you can modify logging configuration parameters through the COREid system console, providing that the AUTOSYNC parameter in the configuration file has previously been set to the default value True. (See “Configuring Logs in the COREid System Console” on page 390).

About Embedded Comments

As installed, each log configuration file contains extensive embedded comments that explain the parameters you set to control log output. Comments, which can span one or multiple lines, begin with a left angle-bracket, an exclamation point, and two dashes, followed by two blank spaces (“<!-- ”). They end with two spaces followed by two dashes, an exclamation point, and a closing angle-bracket (“ --!>”).

When you use the COREid System Console to modify the log parameters for a component, then commit those changes, the configuration file associated with that component is recorded to disk without the embedded comments. The presence or absence of these comments does not affect logging in any way; they are included solely to guide manual editing of the log configuration file.

In any case, you can view the original comments by opening the read-only duplicate of the original logging configuration file, which is named “oblog_config_original.xml” and located in the following directory:

```
Component_install_dir/oblix/config
```

where *Component_install_dir* is the root installation folder for your COREid Server.

The following listing presents a typical log configuration file with comments embedded. For an example of a log file without embedded comments, see “The Default Log Configuration File (without embedded comments)” on page 388.

Listing 2 The Default Log Configuration File (with Embedded Comments)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--===== -->
<!--===== -->
<!--NetPoint Logging Configuration File -->
```

```

<!-->
<!--For changes to this file to take effect, stop and restart the -->
<!--server affected by the changes. For instance, if you make -->
<!--changes to COREid logging, stop and restart the COREid Server. -->
<!-->
<!-->
<!--===== -->
<!--===== -->
<!--Set the Log Threshold -->
<!-->
<!--The log Threshold determines the amount of information to log. -->
<!--Selecting a lower level of logging includes the information -->
<!--logged at the higher levels. For example, LOGLEVEL_ERROR -->
<!--includes the information collected at LOGLEVEL_FATAL. -->
<!-->
<!--Choices are: -->
<!--LOGLEVEL_FATAL - serious error, possibly a program halt. -->
<!--LOGLEVEL_ERROR - a transient or self-correcting problem. -->
<!--LOGLEVEL_WARNING - a problem that does not cause an error. -->
<!--LOGLEVEL_INFO - reports the current state of the component. -->
<!--LOGLEVEL_DEBUG1 - basic debugging information. -->
<!--LOGLEVEL_DEBUG2 - advanced debugging information. -->
<!--LOGLEVEL_DEBUG3 - logs performance-sensitive code. -->
<!--LOGLEVEL_TRACE - used when you need to trace the code path -->
<!--execution or capture metrics. Includes all previous levels. -->
<!-->
<!--If you do not specify a threshold, the default is WARNING. -->
<!-->
<!--In addition to specifying a threshold, you need to specify -->
<!--if changes that you make to the logging configuration in -->
<!--the NetPoint GUI overwrite the settings in this file. The -->
<!--AutoSync parameter accomplishes this. This parameter takes a -->
<!--value of True or False. If set to True, changes made in the -->
<!--GUI overwrite changes in this config file. If False, changes -->
<!--made in the GUI are only in effect until the server is -->
<!--stopped or restarted, after which the settings in this file -->
<!--overwrite the GUI settings. The default is True. -->
<!-->
<!-->
<CompoundList xmlns="http://www.oblix.com" ListName="logframework.xml.staging">
  <SimpleList>
    <NameValPair ParamName="LOG_THRESHOLD_LEVEL" Value="LOGLEVEL_WARNING" />
    <NameValPair ParamName="AUTOSYNC" Value="True" />
  </SimpleList>
<!-->
<!-->
<!--===== -->
<!--===== -->
<!--Configure the Log Level -->
<!-->
<!-->
<!--To configure a log level, you specify a name for the -->
<!--configuration (for instance, MyErrorLog1) and -->

```

```

<!--the log level that you are configuring. You can create      -->
<!--more than one configuration per log level if you want      -->
<!--to output to more than one destination. You can output to -->
<!--the system log or to a file, as specified on              -->
<!--the LOG_WRITER parameter. The value for the LOG_WRITER    -->
<!--parameter may only be SysLogWriter, FileLogWriter or      -->
<!--MPFileLogWriter. The MPFileLogWriter is a multi-process safe -->
<!--FileLogWriter. It should be used to log in webcomponents i.e -->
<!--WebGate, Access Manager and WebPass loaded on multiprocess -->
<!--webservers like Apache and IPlanet(Unix)                  -->
<!------>
<!--If you do not specify an output destination, the default is -->
<!--SysLogWriter.                                             -->
<!------>
<!--If outputting to a file, you also specify a file name and -->
<!--other parameters. Default parameter values are:          -->
<!--FILE_NAME: <installdir>/oblix/log/oblog.log              -->
<!--BUFFER_SIZE: 32767 (number of bytes)                      -->
<!--MAX_ROTATION_SIZE: 5242880 (bytes, equivalent to 5MB)    -->
<!--MAX_ROTATION_TIME: 86400 (seconds, equivalent to one day) -->
<!------>
<!--Configuring the log level does not ensure that the data is -->
<!--actually collected. Data collection for a log is          -->
<!--determined by the LOG_THRESHOLD_LEVEL parameter, above,   -->
<!--and the LOG_STATUS parameter in the log configuration.    -->
<!------>
<!--If you do not provide a LOG_STATUS, the default for       -->
<!--LOGLEVEL_FATAL, LOGLEVEL_ERROR, and LOGLEVEL_WARNING,    -->
<!--is On.                                                     -->
<!------>
<!--This file contains several sample configurations that are -->
<!--enclosed in comments. To use them, remove the comments.  -->
<!------>
  <CompoundList xmlns="http://www.oblix.com" ListName="LOG_CONFIG">
<!--Write all FATAL logs to the system logger. -->
    <ValNameList xmlns="http://www.oblix.com" ListName="LogFatal2Sys">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_FATAL" />
      <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
<!--Write all ERROR logs to the system logger. -->
    <ValNameList xmlns="http://www.oblix.com" ListName="LogError2Sys">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
      <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
<!--Write all WARNING logs to the system logger. -->
    <ValNameList xmlns="http://www.oblix.com" ListName="LogWarning2Sys">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_WARNING" />
      <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
<!--Write all logs to the Oblix log file. -->

```

```

<ValNameList xmlns="http://www.oblix.com" ListName="LogAll2File">
  <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ALL" />
  <NameValPair ParamName="LOG_WRITER" Value="FileLogWriter" />
  <NameValPair ParamName="FILE_NAME" Value="oblog.log" />
<!--Buffer up to 64 KB (expressed in bytes) of log entries before flushing to the
file. -->
  <NameValPair ParamName="BUFFER_SIZE" Value="65535" />
<!--Rotate the log file once it exceeds 50 MB (expressed in bytes). -->
  <NameValPair ParamName="MAX_ROTATION_SIZE" Value="52428800" />
<!--Rotate the log file after 24 hours (expressed in seconds). -->
  <NameValPair ParamName="MAX_ROTATION_TIME" Value="86400" />
  <NameValPair ParamName="LOG_STATUS" Value="On" />
</ValNameList>
</CompoundList>
</CompoundList>

```

About Log Writers

In addition to controlling the content of component-specific logs (in other words, the levels of logging that are reported), you can send the output collected at any log level to the log “writer” of your choice. For instance, you can direct catastrophic errors to the system log, but send trace-level debugging information to a disk file of your choice.

You determine where log data gets sent by setting the value of the `LOG_WRITER` parameter in a log-handler definition in the log configuration file.

Each of the three log writers supplied by NetPoint formats log data into an appropriate format and directs the output to a specific destination such as a system log or a data file. These log writers are described in Table 53.

Table 53 NetPoint Log Writers

Writer	Description
SysLogWriter	<p>This writer records data to the system log file for the machine that hosts the NetPoint component being logged.</p> <p>For Windows machines, this is the application log file, which you can view by navigating to: My Computer > Manage > Event Viewer > Application.</p> <p>For Unix platforms, the name and location of the system log file can vary according to the machine and the preferences of the system administrator. Consult the “owner” of the machine for exact details on where to find the file.</p> <p>Typically, the system log file contains event information recorded not just by NetPoint, but by other applications and the host operating system as well.</p> <p>By default, the NetPoint logging configuration file specifies that Fatal, Error, and Warning messages be sent to the system file.</p>

Table 53 NetPoint Log Writers

Writer	Description
FileLogWriter	<p>This writer is recommended when you want a disk file to record log data for an Access Server, COREid Server, or other single-process application.</p> <p>This writer enables you to specify the size of the buffer used for writing the file, the size at which the file is rotated, and the interval at which the file is rotated, regardless of size.</p> <p>FileLogWriter opens the log file and holds it open for disk writes until the approximate file size limit or file rotation interval has been reached; therefore, it is unsuitable for situations in which more than one process needs to write to the same log file. For logging in multi-process situations, see MPFileLogWriter, immediately below.</p>
MPFileLogWriter	<p>This writer resembles the FileLogWriter, except it opens and closes the log file each time it writes data to the file. This enables multiple processes to write to the file in turn. However, this practice can slow NetPoint performance substantially. Therefore, Oblix recommends using MPFileLogWriter only when FileLogWriter fails to record logging data from some of the processes associated with a multi-process application such as an AccessGate installed on a multi-process Web server (such as Apache) or the Linux or Solaris versions of the iPlanet Web server.</p>

Log Configuration File Structure

The log configuration file conforms to a standard format, which is parsed during component start-up and at other key points. Although you can edit parameters and add or subtract certain sections known as log-handler definitions, you should not change the underlying format of the log configuration file, or else the configuration parameters may become unparsable.

The following table lists the elements in a NetPoint log configuration file with examples included as well. (The positions of elided content are indicated by ellipses.) For a listing of the default NetPoint log configuration file, see “The Default Log Configuration File (with Embedded Comments)” on page 377 or “The Default Log Configuration File (without embedded comments)” on page 388.

Table 54 Log Configuration File Structure (with Sample Content)

<p>An XML file header that declares the relevant XML version, which is always 1.0, and the encoding format, which is always ISO-8559-1. Note that this header statement differs from the other XML statements in this file in that it begins with “<?” and ends with “?”</p> <pre data-bbox="344 1528 972 1553"><?xml version="1.0" encoding="ISO-8859-1" ?></pre> <p>A compound list that contains:</p> <pre data-bbox="344 1624 843 1649"><CompoundList...>...</CompoundList></pre>
--

Table 54 Log Configuration File Structure (with Sample Content)

The relevant XML name space for the log configuration file (within the opening tag)
`xmlns="http://www.example.com"`

The name of the compound list (within the opening tag)

`ListName="logframework.xml.staging"`

A simple list that contains:

`<SimpleList>...</SimpleList>`

A name/value pair for the LOG_LEVEL_THRESHOLD parameter:

`<NameValPair ParamName="LOG_THRESHOLD_LEVEL"
Value="LOGLEVEL_WARNING" />`

Another name/value pair for the AUTOSYNC parameter:

`<NameValPair ParamName="AUTOSYNC" Value="True" />`

One or more compound lists, which, at this particular level, are known as log-handler definitions. Each contains:

`<CompoundList...>...</CompoundList>`

The relevant XML name space (within the opening tag)

`xmlns="http://www.example.com"`

The name of the compound list (within the opening tag)

`ListName="LOG_CONFIG"`

And one or more value/name lists, each of which contains:

`<ValNameList...>...</ValNameList>`

The relevant XML name space (within the opening tag)

`xmlns="http://www.example.com"`

The name of the value/name list (within the opening tag)

`ValNameList ListName="LogFatal2Sys"`

The following three mandatory name/value pairs:

The LOG_LEVEL parameter

`<NameValPair ParamName="LOG_LEVEL"
Value="LOGLEVEL_FATAL" />`

The LOG_WRITER parameter

`<NameValPair ParamName="LOG_WRITER"
Value="SysLogWriter" />`

Table 54 Log Configuration File Structure (with Sample Content)

<p>The LOG_STATUS parameter</p> <pre><NameValPair ParamName="LOG_STATUS" Value="On" /></pre> <p>And none, some, or all of the following four name/value pairs, which are relevant only if you specified FileLogWriter or MPFileLogWriter for the LOG_WRITER parameter.:</p> <p>The FILE_NAME parameter</p> <pre><NameValPair ParamName="FILE_NAME" Value="oblog.log" /></pre> <p>The BUFFER_SIZE parameter</p> <pre><NameValPair ParamName="BUFFER_SIZE" Value="65535" /></pre> <p>The MAX_ROTATION_SIZE parameter</p> <pre><NameValPair ParamName="MAX_ROTATION_SIZE" Value="52428800" /></pre> <p>The MAX_ROTATION_TIME parameter</p> <pre><NameValPair ParamName="MAX_ROTATION_TIME" Value="86400" /></pre>
--

About XML Element Order

The XML tag language employs a tree-like structure with lists of elements corresponding to the leaves on a branch.

Within a given list, parallel elements can be presented in any order as long as the elements themselves remain intact and entirely within the tags that originally bracketed them. For example, the following two name/value lists are equivalent:

```
<ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
  <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
  <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
  <NameValPair ParamName="LOG_STATUS" Value="On" />
</ValNameList>
```

```
<ValNameList xmlns="http://www.example.com" ListName="LogError2Sys">
  <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
  <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
  <NameValPair ParamName="LOG_STATUS" Value="On" />
</ValNameList>
```

Similarly, within a given tag, the attributes (except for the tag name, which must always be the first element within the tag brackets) can be reordered, as long as they remain intact and within the tag elements that originally bracketed them. For example, the following two opening tags for a name-value list are equivalent:

```
<ValNameList xmlns="http://
www.example.com"
ListName="LogError2Sys">
```

```
<ValNameList ListName="LogError2Sys"
xmlns="http://www.example.com">
```

Controlling Logging Levels

Up to four interconnected factors determine whether logging is active for a given NetPoint component at a given log level. These factors are listed in the following table:

Table 55 Factors that Determine Whether Logging Is Active

Factor	Importance	Description
LOG_THRESHOLD_LEVEL	Primary	This parameter provides a convenient means to limit log output through a single setting. It takes precedence over all other settings by setting an absolute threshold within the log level hierarchy described in Table 51, "NetPoint Logging Levels," on page 374. Below the threshold level, no logging can take place, regardless of the other settings. For COREid Servers only, see "Configuring Logs in the COREid System Console" on page 390 for details on the relationship between configuration file and GUI-based settings.
LOG_STATUS	Secondary	This parameter toggles logging on or off, providing it does not get overridden by the log threshold level. See the previous row for details.
AUTOSYNC	Secondary	When this parameter is set to True, you can save log settings in the COREid System Console to the log configuration file. When AUTOSYNC is False, and you click Save on either the Log Handler Definition or Modify COREid Server pages, the changes take effect on the server, but they do not get written to the log configuration file, and they do not persist after you restart the server.
The physical position of a log handler	Secondary	See "About Log Handler Precedence" on page 385.

About Log Handler Precedence

A single log-configuration file can contain as many as three log-handler definitions for a single log level. You can have this many, because three different log handlers are required if you wish to send output to each of the three log writers.

When the LOG_STATUS settings in these log handlers conflict, the setting in the log-handler definition closest to the physical end of the log configuration file is read last, after the associated NetPoint component is restarted. Therefore, it takes precedence over the LOG_STATUS settings in all previous log-handler definitions for the same log level.

Because the state of the LOG_STATUS parameter in the “last read” log-handler definition for a given level takes effect for *all* the log-handler definitions for that level, it is therefore possible to set LOG_STATUS to Off for the first two log handlers that aim at a certain level, yet logging can still occur for all three handlers, because LOG_STATUS happens to be On for the third and final log handler in the configuration file.

As previously mentioned, the LOG_STATUS settings at any given level become moot if that level lies below the current LOG_THRESHOLD_LEVEL. In such a case, neither conflicting settings among the log handlers, nor the order in which the log handlers appear is of consequence, because logging cannot be activated at this level.

Ensuring That Your Edits Take Effect

To make your manual edits to a logging configuration file take effect, stop and restart the NetPoint component associated with that configuration file.

Note: For COREid Servers, edits made through the GUI are written to oblog_config.xml and take effect only if “AutoSync” is set to “True.”

Log Configuration Parameters

At minimum, each log-handler definition sets five parameters, as listed in Table 56:

Table 56 Mandatory Log Configuration File Parameters

Parameter	Comment
xmlns	This specifies the relevant XML namespace for the current list and is identical for all log-handler definitions in a given logging configuration file. Example: <code>http://www.example.com</code>
ListName	These names are required for all the lists in the logging configuration file. Wherever possible, preserve the default list names. When creating a new log-handler definition, try to select a name for the associated name/value list that easily distinguishes the entry from all other entries in the logging configuration file. Examples: <code>WarningsAndAboveToSyslog</code> sends Fatal, Error, and Warning messages to the system log file. <code>WarningsOnlyToFileLog128KBuffer</code> sends messages from just the Warning level to a 128KB buffer, and hence to a disk file. <code>TraceOnlyToMPRotateDaily</code> sends messages from just the Trace level to the multi-process file writer, which opens and closes the file each time it writes to disk. This file is replaced with a fresh (empty) file every day, regardless of the size of the file at the time of replacement.
LOG_LEVEL	This specifies one of the nine available log level settings. See the table “NetPoint Logging Levels” on page 374. The default logging configuration file activates logging for three levels: Fatal, Error, and Warning. Output is sent to both the system log and the log data file for the NetPoint component doing the logging.
LOG_WRITER	This specifies which log writer handles output for a given log-handler definition. See “NetPoint Log Writers” on page 380 for a list of the supported choices.
LOG_STATUS	This parameter turns the log handler on or off, as explained in the next section.

If you specify `FileLogWriter` or `MPFileLogWriter` as for the `LOG_WRITER` parameter, the four parameters detailed in the following table become relevant. The first becomes mandatory, while the other three are optional.

Table 57 Log Data File Configuration Parameters

Parameter	Description	Default
FILE_NAME	<p>Used only for the FileLogWriter or MPFileLogWriter. It represents the name (and location) of the file to which logging information is written.</p> <p>You can prepend an absolute path to the file name so as to store it somewhere other than the default location, which is:</p> <p><i>Component_install_dir</i>\oblix\logs where <i>Component_install_dir</i> is the root installation directory for the component whose system events you are logging.</p> <p>If you do not specify a file name, the default applies.</p> <p>When you create more than one log-handler definition that sends output either to FileLogWriter or MPFileLogWriter, make sure that you specify different file names in each case so that multiple handlers do not attempt to write to the same file. This caution does not apply to log handlers accessing the SysLogWriter.</p>	oblog.log
BUFFER_SIZE	<p>This parameter represents the size of the buffer used to store logged data being written to the log file.</p> <p>If you set the buffer value to 0, no buffering is performed. (This ability to turn off buffering can be useful when a crash occurs).</p> <p>In the event of a crash, Fatal-level messages are always flushed to the log file.</p> <p>If you do not specify the buffer size, the default applies.</p>	65535 (64KB)
MAX_ROTATION_SIZE	<p>When the log file reaches this size (in bytes), the file is renamed and a new file is created with the file name originally used by the just-renamed file. For example "oblog.log" becomes "oblog.log 1081303126." The number represents the time when the file was created.</p> <p>If you do not specify this parameter, the default is used.</p>	52428800 (512KB)
MAX_ROTATION_TIME	<p>The time interval, in seconds, when the log file is renamed, whether or not it has reached the maximum rotation size. If the maximum log file size is not reach between two time-triggered file rotations, the numbers appended to the log files created differ by the number of seconds in the rotation interval. For example, "oblog.log. 1081389526" and "oblog.log. 1081303126" differ by 84.600, which is the number of seconds in 24 hours, the rotation interval set in the logging configuration file.</p> <p>If you do not specify this parameter, the default is used.</p>	86400 (1 day, in seconds)

Default Log Settings

The default log configuration file installed with each NetPoint component activates only the highest three levels (Fatal, Error, and Warning) in the hierarchy of logged events.

Also by default, all log output is directed to the system log.

On Windows machines, you can view the system log for the machine hosting the NetPoint component you are logging by navigating to: My Computer > Manage > Event Viewer > Application. System event entries for the NetPoint components being logged are interspersed among the system events reported for the operating system and non-NetPoint applications.

For the Solaris and Linux environments, the location of the system log is recorded in a system configuration file whose particulars can vary from machine to machine. For the name and location of this system file, consult the owner of the machine hosting the NetPoint component whose system log you wish to examine.

The following listing presents the content of the default log-configuration file installed with each NetPoint component. The embedded comments, which have no effect on the actual function of the file, have been removed in order to expose the underlying structure of the file:

Listing 3 The Default Log Configuration File (without embedded comments)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<CompoundList xmlns="http://www.example.com"
  ListName="logframework.xml.staging">
  <SimpleList>
    <NameValPair ParamName="LOG_THRESHOLD_LEVEL"
      Value="LOGLEVEL_WARNING" />
    <NameValPair ParamName="AUTOSYNC" Value="True" />
  </SimpleList>
  <CompoundList xmlns="http://www.example.com" ListName="LOG_CONFIG">
    <ValNameList xmlns="http://www.example.com"
      ListName="LogFatal2Sys">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_FATAL" />
      <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
    <ValNameList xmlns="http://www.example.com"
      ListName="LogError2Sys">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ERROR" />
      <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
      <NameValPair ParamName="LOG_STATUS" Value="On" />
    </ValNameList>
    <ValNameList xmlns="http://www.example.com"
      ListName="LogWarning2Sys">
      <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_WARNING" />
      <NameValPair ParamName="LOG_WRITER" Value="SysLogWriter" />
    </ValNameList>
  </CompoundList>
</CompoundList>
```

```

    <NameValPair ParamName="LOG_STATUS" Value="On" />
</ValNameList>
<ValNameList xmlns="http://www.example.com" ListName="LogAll2File">
  <NameValPair ParamName="LOG_LEVEL" Value="LOGLEVEL_ALL" />
  <NameValPair ParamName="LOG_WRITER" Value="FileLogWriter" />
  <NameValPair ParamName="FILE_NAME" Value="oblog.log" />
  <NameValPair ParamName="BUFFER_SIZE" Value="65535" />
  <NameValPair ParamName="MAX_ROTATION_SIZE" Value="52428800" />
  <NameValPair ParamName="MAX_ROTATION_TIME" Value="86400" />
  <NameValPair ParamName="LOG_STATUS" Value="On" />
</ValNameList>
</CompoundList>
</CompoundList>

```

Parsing the Default Log Configuration File

The default log configuration file follows the abstract structure presented in “Log Configuration File Structure” on page 381.

The simple list near the top of the file sets `LOG_THRESHOLD_LEVEL` to the Warning level. Since this parameter takes precedence over all others, none of the levels below Warning are logged, regardless of the settings in the rest of this file.

The simple list also sets the `AUTOSYNC` parameter to True. This setting enables you to save the configuration values you set in the COREid system console to this configuration file so that they “persist,” even after you restart the COREid Server. Although the `AUTOSYNC` setting appears in the default configuration files for all of the NetPoint components, it is relevant only for COREid Servers.

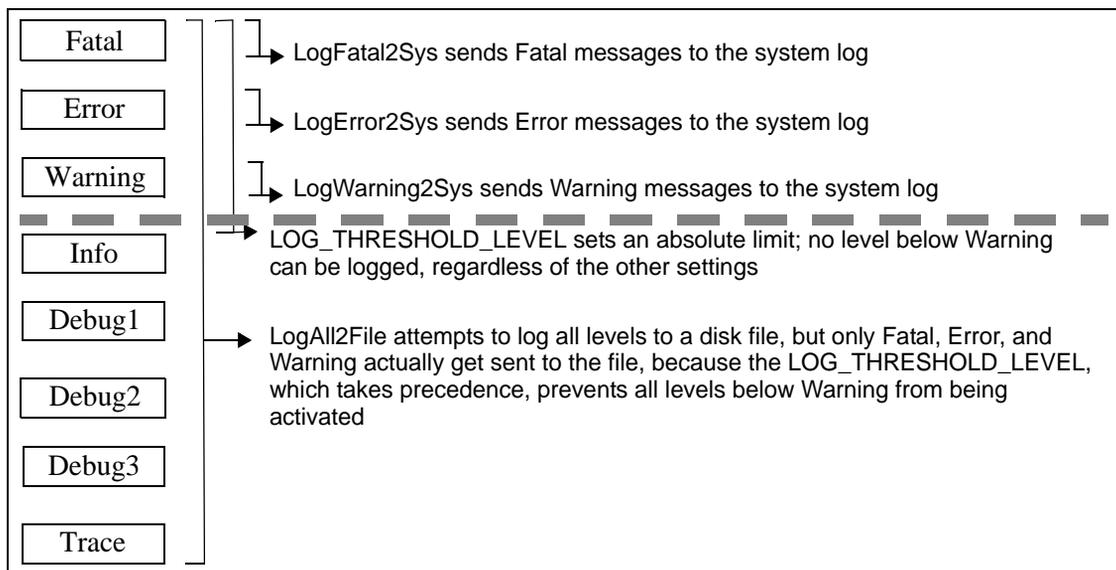
The nested compound list contains four log-handler definitions. The first, which is named, `LogFatal2Sys`, sets the `LOG_LEVEL` affected by this definition to the Fatal level and sets `LOG_STATUS` to On. As previously noted, the threshold level for this configuration file is Warning, which is below Fatal, so this definition is not overridden. The log output goes to the system log, because this is what the definition specifies through the `LOG_WRITER` parameter.

The `LogError2Sys` log-handler definition sends Error level messages to the system log. Error stands above the current threshold level (Warning), so this definition is in effect.

The `LogWarning2Sys` definition sends Warning level output to the system log. Like the two previous log-handler definitions, it is not overridden by the current `LOG_THRESHOLD_LEVEL` parameter.

`LogAll2File`, the final log-handler definition, appears to send output from all eight log levels to a disk file named `oblog.log`. However, `LOG_THRESHOLD_LEVEL`, which is currently set to Warning, takes precedence, so only the output from the Fatal, Error, and Warning levels gets recorded in the log file.

Figure 13 Log-Level Activation in the Default Log Configuration File



Since output from LogAll2File goes to the FileLogWriter, the parameters governing file name, buffer size, rotation size, and rotation interval all take effect.

In sum, the default configuration file sends Fatal, Error, and Warning messages to both the system log and a default log data file named oblog.log.

Configuring Logs in the COREid System Console

For COREid Servers only, you can modify certain log settings through the COREid System Console. Alternatively, you can edit the log configuration file manually.

To view or modify log-handler definitions

1. From the COREid System Console, click System Admin > System Configuration > Configure COREid Server.

The List All COREid Servers page appears.



2. Click the link for the COREid Server whose activity you want to log.

The Details for COREid Server page appears with a list of log-handler definitions at the bottom of the page.

Details for COREid Server

Name	np70-COREid1-7010									
Hostname	brass									
Port	7010									
Debug	Off									
Debug File Name	/oblix/logs/debugfile.lst									
Transport Security	Simple									
Maximum Session Time (hours)	24									
Number of Threads	20									
Audit to Database Flag (auditing on/off)	Off									
Audit to File Flag (auditing on/off)	Off									
Audit File Name										
Audit File Maximum Size (bytes)	100000									
Audit File Rotation Interval (seconds)	7200									
Audit Buffer Maximum Size (bytes)	25000									
Audit Buffer Flush Interval (seconds)	7200									
Log Level	Warning and above									
Log Handler Definitions	<table> <thead> <tr> <th>Name</th> <th>Log Level</th> <th>Output To</th> </tr> </thead> <tbody> <tr> <td>LogFatal2Sys</td> <td>Fatal</td> <td>System Log</td> </tr> <tr> <td>LogAll2File</td> <td>All Log Levels</td> <td>File</td> </tr> </tbody> </table>	Name	Log Level	Output To	LogFatal2Sys	Fatal	System Log	LogAll2File	All Log Levels	File
Name	Log Level	Output To								
LogFatal2Sys	Fatal	System Log								
LogAll2File	All Log Levels	File								
Scope File Name	/oblix/logs/scopefile.lst									
SNMP State	Off									
SNMP Agent Registration Port										
Directory Server Profiles Used	default-np70-COREid1-7010									

3. Examine the Log Level field above the Log Handler Definitions table. This represents the current LOG_THRESHOLD_LEVEL. If you want to change this setting, click Modify at the bottom of the page and proceed to “To modify the log threshold from the COREid System Console” on page 392. Otherwise, continue to the next step.
4. In the table of log-handler definitions, click the link for the log handler you wish to examine or change.

The Modify Log Writer page appears. From this page, you can specify values as described in “Mandatory Log Configuration File Parameters” on page 386. If you specify File in the Output field, you must complete the fields described in “Log Data File Configuration Parameters” on page 387.

You can change the defaults for buffer size, maximum log file size, default file rotation interval, and file name, which are listed in “Log Data File Configuration Parameters” on page 387.

Modify the Log Handler Definition.

Name*

Log Level*

Output To* System Log
 File

Log File Name

Log File Maximum Size (bytes, 5242880 bytes equals 5 MB)

Log File Rotation Interval (seconds, 86400 seconds equals one day)

Log Buffer Maximum Size (bytes)

Note: The fields marked with an asterisk(*) are required fields.

To modify the log threshold from the COREid System Console

1. If the Details of the COREid Server page is not already displayed, navigate to COREid System Console, click System Admin > System Configuration > Configure COREid Server, then click the name of the COREid Server whose settings you want to examine.
2. Click Modify at the bottom of the Details of the COREid Server page.
3. Use the drop-down list to set the Log Threshold Level to the value you wish.
4. Click Save at the bottom of the page to make the change take place on the server

If AUTOSYNC is True in the log configuration file, the change is written to the log configuration file so that the change persists after you restart the server.

To add or delete a log-handler definition

1. From the COREid System Console, click System Admin > System Configuration > Configure COREid Server, then click the name of the COREid Server to which you wish to add a log-handler definition.
2. Click Modify at the bottom of the page.

The Modify COREid Server page appears.

3. Locate the Log Writers field and complete the appropriate action below:
 - To delete a log output configuration, check the box next to the appropriate link, then click Delete.
 - To add a log writer, click Add.

If you click Add, the Add a New Log Writer page appears.

4. Supply a name and a log level for the new log writer.
5. Verify that the log level is the same as or higher than the current log threshold level, as determined in step 3 in the procedure “To view or modify log-handler definitions” on page 390.

If the new log level is lower than the current threshold level, set the threshold level to the new log level or lower, as detailed in “To modify the log threshold from the COREid System Console” on page 392.

6. If you choose to output to a file rather than the system log, you must supply a file name and path, as described in Table 57, “Log Data File Configuration Parameters,” on page 387.

11 Auditing

This chapter focuses on all aspects of the NetPoint auditing feature, which it details through the following topics:

- “About NetPoint Auditing” on page 395
- “Audit Output Considerations” on page 396
- “Controlling Audit Output” on page 399
- “NetPoint Auditing Requirements” on page 403
- “Audit-to-Database Architecture” on page 405
- “Setting Up File-Based Auditing” on page 412
- “Setting Up Database Auditing” on page 416
- “Setting up NetPoint Audit Reports” on page 451

About NetPoint Auditing

The NetPoint auditing feature collects and presents data pertaining to NetPoint policy and profile settings, system events, and usage patterns. NetPoint can generate two broad types of audit reports:

- **Static**—These reports are derived from policy and profile information stored on the NetPoint directory server. For details, see “Static Audit Reports” on page 397.
- **Dynamic**—These reports are derived from Access System and COREid System event information collected from the NetPoint servers in your system. At the most detailed level, dynamic audit reports reveal when a system event was triggered and who triggered it. At a higher level, these reports can reveal component load levels, resource request patterns, system intrusion attempts, and overall NetPoint system performance. For details, see “Static Audit Reports” on page 397.

Auditing stands as one of many NetPoint features that collect and present NetPoint-related information. For an overview of NetPoint Logging, SNMP Monitoring, and other reporting features, see “NetPoint Reporting” on page 367.

Audit Output Considerations

You can record all dynamic audit reports and some static audit reports to disk file, to a relational database, to both destinations, or to neither. Additionally, some static reports can be displayed in limited form through the NetPoint graphical user interface.

Audit Security Considerations

Database auditing provides the following advantages in the area of security:

- All audit information is stored in a central database, which can be protected by whatever security methods your relational database application supports. By contrast, the audit-to-file option records data to a plain-text file on each server that collects audit data. Such files are not protected by database-level security.

Important: To take full advantage of database security, make sure you turn off the audit-to-file feature for all Access and COREid Servers in your NetPoint system. You should also store the password to the default audit database user account in the RDBMS profile on the Idirectory server, rather than in the ODBC.ini file on each NetPoint server host.

- Data can be sent to the audit database using the transport security methods supported by ODBC.
- Using the audit database, Crystal Reports can generate security-related statistics. For instance, you can track the number of resource requests refused during a given interval or compile a list of users locked out of the system.
- Auditing-to-database can assist in compliance reporting for regulatory acts such as Sarbanes-Oxley, Gramm-Leach-Bliley, and HIPAA (the Health Information Privacy and Accountability Act of 1996).

Audit Performance Considerations

Auditing, whether to database or file, can slow the performance of your NetPoint System. You can control the impact of auditing by modifying the following parameters:

- Turn on auditing only for selected servers. (See “To enable and configure auditing for each COREid Server” on page 439 and “To enable and configure auditing for each Access Server” on page 447.)

- Turn on auditing only for selected profile attributes, events, and COREid applications. (See “To specify global COREid system events and profile attributes for audit” on page 441, “To specify User Manager events for audit” on page 442, “To specify Group Manager events for audit” on page 443, and “To specify Organization Manager events for audit” on page 444.)
- Increase the “DB audit retry” interval so that whenever the connection to the database is broken, the NetPoint server does not initiate “thrashing” by resending a failed write attempt before the connection is restored. You change this setting by manually editing the DBAuditRetryInterval parameter in the globalparams.lst file, which is located in the following directory:

Component_install_dir\apps\common\bin\

where *Component_install_dir* is the installation directory of the NetPoint server whose audit behavior you wish to control.

Use any plain text editor to change the number of seconds to wait before initiating another attempt to write data to the audit database.

- For file-based auditing only, increase the size of the audit buffer. This measure reduces the number of times the audit feature accesses your hard disk. (See “To enable and configure auditing for each COREid Server” on page 439 and “To enable and configure auditing for each Access Server” on page 447.)

Important: Only fatal errors are flushed to file if your NetPoint server crashes. All other audit items in the buffer at the moment of the crash are lost. Therefore, by increasing the buffer size or lengthening the interval between buffer flushes, you increase the potential volume of audit data lost in the event of a crash.

- For file-based auditing only, lengthen the interval between buffer flushes. This reduces the number of times the auditing feature writes to disk, but it also increases the potential amount data you lose in a crash. See the preceding cautionary note.

Static Audit Reports

Static audit reports are generated from policy and profile information stored on the NetPoint directory server. You can generate five types of static reports:

Table 58 Static Audit Report Types

Report Type	Description
User Access Privilege Report	A global list of resources that a specified user or group of users can access at a specified point in time. They are also referred to as filtered profile queries. (For details, see the procedure “To create and manage User access privilege reports” on page 449.)

Table 58 Static Audit Report Types

Report Type	Description
Resource Access Privilege Report	A global list of users authorized to access a specified resource or group of resources at a specified point in time. They are also referred to as filtered policy queries. (For details, see the procedure “To create and manage User access privilege reports” on page 449.)
Access Test	A limited, on-screen display that verifies whether a specified user or group of users can access a specified resource at a specified point in time. (You cannot test for access to randomly defined groups of resources the way you can with the preceding two types of filtered queries.)
Access System Diagnostic Report	An on-screen table containing status information on some or all of the Access Servers in your system. This includes details about the directory components to which the Access Server(s) are connected. (For details, see: “Where to Set Specific NetPoint Audit Options” on page 400.)
COREid System Diagnostic Report	An on-screen table containing status information on some or all of the COREid Servers in your system. This includes details about the directory components to which the COREid Server(s) are connected. (For details, see: “Where to Set Specific NetPoint Audit Options” on page 400.)

Dynamic Audit Reports

If you send data to the audit database, you must install and configure one of the following databases on a host within your NetPoint domain:

- Microsoft SQL Server for environments in which all the machines hosting NetPoint servers are running Windows. (For details, see “About installing SQL Server (Windows)” on page 418.)
- MySQL for Unix for environments in which all the machines hosting NetPoint Servers are running Unix. For details, see “About installing MySQL (Unix)” on page 419.

In addition, you must install and configure Crystal Reports (and a required patch) on a Windows machine within your NetPoint domain. Crystal Reports is bundled with NetPoint. (For details, see “To install Crystal Reports” on page 451.)

If you use MySQL for Unix as the server for your audit database, you must install and configure the MyODBC database driver on all machines hosting NetPoint Servers that connect to your MySQL database. For details, see “Task overview: To install the MyODBC driver (Unix only)” on page 429.

Note: Currently, Solaris 8 and Solaris 9 are the only Unix variants tested for use with the NetPoint audit-to-database feature. Therefore, only the Solaris-specific versions of MySQL and MyODBC can be deployed in conjunction with the NetPoint audit-to-database feature. See “To obtain the MySQL installation package” on page 420 and “To obtain and unpack the installation package for MyODBC” on page 429 for the specific MySQL and MyODBC packages against which NetPoint 7 has been tested.

If you use SQL Server for your audit database, you do not have to install any database drivers explicitly, because they are installed by default as part of your Windows environment.

Controlling Audit Output

You can control the type and amount of audit data collected by each NetPoint server.

For example, you can configure the Master Audit Rule on an Access Server to record authentication failures, but not authentication successes. (For details, see “To modify audit output formatting for the Access system” on page 448.)

Similarly, you can configure the Application Auditing Policy on a COREid Server to record the time and date of each user logon, but not the time of logon or session expiration. (For details, see “To modify audit output formatting for the COREid system” on page 440.)

If you send data to the audit database, you can display the collected information in Crystal Reports templates that have been pre-configured to present NetPoint audit data. The generated reports are known as NetPoint Audit Reports, which fall into the following broad categories:

- Global View Access
- Authentication
- Authorization
- Activity
- Identity Management

For details see “About NetPoint Audit Reports” on page 411.

About NetPoint Audit Options

You set all audit options through various pages in the NetPoint graphical user interface, as detailed in Table 59:

Table 59 Where to Set Specific NetPoint Audit Options

Audit-Related Functionality	Location in GUI	Scope
<p>Enable file-based and/or database auditing and modify audit file attributes on an individual COREid Server.</p>	<p>COREid System Console > System Administration > System Configuration > Configure COREid Server > <i>ServerName</i> > Modify</p> <p>where <i>ServerName</i> specifies the COREid Server you want to modify</p>	<p>Per server</p>
<p>Modify the default formatting used for both file-based and database auditing, including date format, date separator, message format, escape character, record separator, and field separator.</p> <p>To enable database auditing, you must replace the default message format string. (See page 440.)</p> <p>If you modify any other attributes, you may have to reconfigure your Crystal Report templates and repository settings.</p>	<p>COREid System Console > Common Configuration > Configure Master Audit Policy > Modify</p>	<p>Global (for both file-based and database auditing within the COREid System)</p>
<p>Specify the COREid system events to be audited. This includes: Success and failure reporting for Login, Logout, Password Management, and Licenses.</p>	<p>COREid System Console > Common Configuration > Configure Global Audit Policies > Modify</p>	<p>Global (for both file-based and database auditing and for all applications within the COREid System)</p>

Table 59 Where to Set Specific NetPoint Audit Options

Audit-Related Functionality	Location in GUI	Scope
<p>Create or modify RDBMS profiles and associated database instances. (These are necessary only for database auditing.)</p>	<p>COREid System Console > System Administration > System Configuration > Configure Directory Options > Configure RDBMS Profiles > Modify</p> <p>or</p> <p>Access System Console > System Configuration > View Server Settings > Configure RDBMS Profiles > Modify</p>	<p>Global (for database auditing only)</p>
<p>You can specify the COREid Servers to be included in the on-screen diagnostics display.</p> <p>Note: To ensure that Diagnostics displays the current status of a given NetPoint Server, “exercise” the connection to that server by attempting a login or a user search before accessing the Diagnostics display.</p>	<p>COREid System Console > System Administration > System Management > Diagnostics</p>	<p>Global (for the COREid system only) or per server</p>
<p>Activate the collection of audit success and/or audit failure data for the following events: Search, View Profile, Modify Profile, View Location, Modify Location, Substitute Right, Workflow, Configuration, Deactivated User, Reactivated User, Created User, Deleted User, and Workflow Duration.</p>	<p>COREid System Console > User Manager Configuration > Configure Auditing Policy > Modify</p>	<p>Global (for User Manager reports only)</p>
<p>Activate the collection of audit success and/or audit failure data for the following events: Search, View Profile, Modify Profile, View My Group, View Group Member, Expand Group, Subscribe Group, Workflow, Configuration, and Workflow Duration.</p>	<p>COREid System Console > Group Manager Configuration > Configure Auditing Policy > Modify</p>	<p>Global (for Group Manager reports only)</p>

Table 59 Where to Set Specific NetPoint Audit Options

Audit-Related Functionality	Location in GUI	Scope
<p>Activate the collection of audit success and/or audit failure data for the following events: Search, View Profile, Modify Profile, Containment Profile, Container Limit, View Location, Modify Location, Workflow, Configuration, and Workflow Duration.</p>	<p>COREid System Console > Org. Manager Configuration > Configure Auditing Policy</p>	<p>Global (for Organization Manager reports only)</p>
<p>Enable file-based and or database auditing and modify audit file attributes on an individual Access Server.</p>	<p>Access System Console > Access System Configuration > Access Server Configuration > <i>ServerName</i> > Modify</p> <p>where <i>ServerName</i> specifies the Access Server you want to modify.</p>	<p>Per server</p>
<p>Create or modify RDBMS profiles and associated database instances. (These are necessary only for database auditing.)</p>	<p>Access System Console > System Configuration > View Server Settings > Configure RDBMS Profiles > Create (or Modify)</p> <p>or</p> <p>COREid System Console > System Administration > System Configuration > Configure Directory Options > Configure RDBMS Profiles</p>	<p>Global (for both file-based and database auditing)</p>
<p>Create or modify a master audit rule, which covers the following: audit events (authentication success and failure as well as authorization success and failure), audit event mapping, date format, escape character, audit record format, and cache formatting.</p> <p>To enable database auditing, you must replace the default audit record format string. (See page 448.)</p> <p>If you modify any other attributes, you may have to reconfigure your Crystal Report templates and repository settings.</p>	<p>Access System Console > Access System Configuration > Common Info Configuration > Master Auditing Rule > Modify</p>	<p>Global (for both file-based and database auditing within the Access System)</p>

Table 59 Where to Set Specific NetPoint Audit Options

Audit-Related Functionality	Location in GUI	Scope
You can specify the Access Server(s) to be included in the on-screen diagnostics display. Note: To ensure that Diagnostics displays the current status of a given NetPoint Server, “exercise” the connection to that server by attempting a login or a user search before accessing the Diagnostics display.	Access System Console > System Management > Diagnostics	Global (for the Access system only) or per server
Create, modify and manage Global User Access Privilege Reports.	Access System Console > System Management > Manage Reports > Add or Modify	Per server

NetPoint Auditing Requirements

The NetPoint auditing requirements depend on the option you select. In general, displaying audit reports on-screen or sending audit output to disk files does not require the installation of special components. By contrast, auditing to database is restricted to certain NetPoint system configurations and also requires the installation of special components, as detailed in the following sections.

Audit-to-Database Requirements

The audit-to-database feature is supported only for NetPoint domains in which all the machines hosting NetPoint servers connected to the audit database are running Windows, or all of them are running Unix. For details, see “Special Components for Database Auditing” on page 404 and “Setting Up Your NetPoint System for Database Auditing” on page 417.

Additionally, database auditing requires that you augment your NetPoint system with special components, as detailed in the next section.

Special Components for Database Auditing

To enable the audit-to-database feature, you must supplement your NetPoint system by installing the components listed in the following table:

Table 60 Special Components Needed for NetPoint Database Auditing

Component	Installation Notes
NetPoint server hosts	<p>All the machines hosting NetPoint servers connected to the audit database must run Windows Advanced Server 2000 SP4 or Windows Server 2003, Enterprise Edition.</p> <p>Alternatively, all the machines hosting NetPoint servers connected to the audit database must run Solaris 8 or Solaris 9. For the latest list of Unix variants supported by the NetPoint auditing feature, consult the NetPoint 7 release notes.</p>
Database server	<p>You must install one of the following database server applications on a machine within your NetPoint domain:</p> <ul style="list-style-type: none"> • Microsoft SQL Server 2000, Standard, Enterprise, or Developer edition for NetPoint environments where all the NetPoint servers connected to the audit database are running on Windows hosts. • MySQL Server 3.5 for NetPoint environments, where all the NetPoint servers connected to the audit database are running on Unix hosts. The NetPoint audit-to-database feature was tested against the following package: <code>mysql-standard-4.0.20-sun-solaris2.8-sparc</code>
Crystal Reports	<p>This reporting software comes bundled with NetPoint 7. You must install Crystal Reports (plus a required patch) on a Windows machine that can access the ODBC database. For details, see “Setting up NetPoint Audit Reports” on page 451. (The Crystal Reports host must run Windows, even in situations where all NetPoint server hosts must run Unix.)</p> <p>The NetPoint audit-to-database feature has been tested against the following Crystal Reports packages: <code>Crystal Reports 9.22a, Advanced Edition</code> <code>patch = CR90DBEXWIN_EN_200403</code></p>
ODBC Drivers	<p>If you use SQL Server, you do not have to install additional database drivers, because Windows already incorporates the drivers necessary to connect your NetPoint servers to your audit database.</p> <p>If you use MySQL, you have to install the MyODBC database driver on each machine hosting a NetPoint server that connects to the audit database. <code>MyODBC-3.51.06-sun-solaris2.8-sparc</code></p> <p>You do not need to install any database drivers for the ODBC and .mdb databases accessed by Crystal Reports, because they are installed by default as part of your Crystal Reports application.</p>

When you install the Solaris version of the Access Server or COREid Server, an ODBC driver manager is installed on the host machine in the following location:

```
Component_install_dir/oblix/lib
```

where *Component_install_dir* is the installation directory for your Access Server or COREid Server.

You do not need to configure this driver manager in any way.

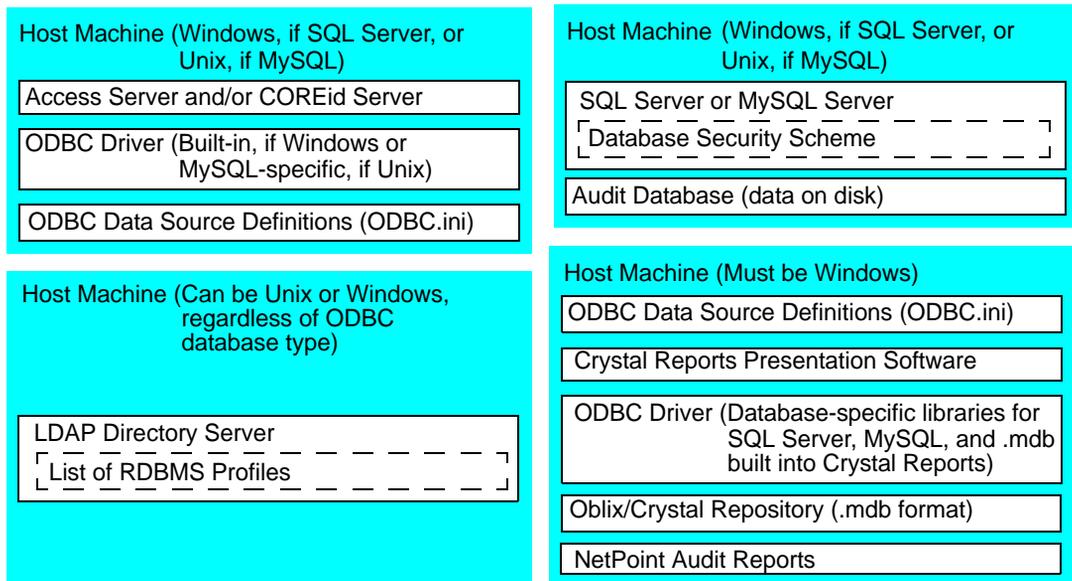
On Windows systems, you do not need to install an ODBC driver manager, because it is an integrated component of the standard Windows installation.

Audit-to-Database Architecture

The following diagram depicts the major components you must install and configure to enable auditing to database:

Figure 14 Audit-to-Database Architecture

SQL Server environments require Windows for all machines hosting NetPoint servers or SQL Server
MySQL environments require Unix for all machines hosting NetPoint servers or MySQL



The preceding diagram assumes that you have installed your NetPoint servers on one or more host machines, your ODBC-compatible database server on another host, and the Crystal Reports application on yet another host.

At the opposite extreme from the preceding “distributed” installation, it is technically possible to install your entire NetPoint system and all the audit-to-database components on a single Windows host, providing that SQL Server is the host for your audit database. (If you use MySQL Server, you cannot place all the components on a single machine, because NetPoint supports only the Unix version of MySQL, and Crystal Reports is a Windows-only application.)

In a single-host scenario, you need only one table of ODBC data-source definitions (one ODBC.ini file) on your host.

About ODBC Data Source Definitions

ODBC data-source definitions encapsulate all the information necessary for a client application such as a NetPoint server or Crystal Reports to connect with ODBC 3.0-compatible databases formatted for SQL Server, MySQL, or Microsoft Access (.mdb).

ODBC data-source definitions are stored in a file named ODBC.ini on each machine that hosts an application connected to the audit database. Only one such list of ODBC data-source definitions should exist on a given machine, and that file should be shared by all the applications that connect to the audit database.

- **Windows**—Users generally add or modify data source definitions through a Windows administration tool GUI. Because this GUI “hides” many configuration details, users may never become explicitly aware that ODBC.ini exists, much less learn its location.
- **Unix**—Users must explicitly set the environment variable ODBCINI so that the database client can locate the ODBC.ini file. Furthermore, they must manually enter or edit the data source definitions in ODBC.ini.

The following table lists the most important attributes in a data source definition:

Table 61 Key Attributes in an ODBC Data-Source Definition

Attribute	Description
DSN (Data Source Name)	<p>Identifies a unique data-source definition to all the clients that access a given data source. (The term DSN is often used incorrectly to denote an entire ODBC data-source definition.)</p> <p>A DSN must be unique within your NetPoint environment. Furthermore, all the ODBC.ini files and RDBMS profiles referencing a particular DSN must contain identical information related to that DSN, including login name, password, database, and so on.</p>

Table 61 Key Attributes in an ODBC Data-Source Definition

Attribute	Description
User	<p>Identifies the database user account authorized to access and modify the ODBC data source. When a NetPoint server or the Crystal Reports application needs to access the data source, it uses this account to supply credentials to the database security scheme.</p> <p>For SQL Server, the default user account is "sa," which stands for system administrator. For MySQL, the default user account is "mysql," which must be a member of the "mysql" group.</p>
Password	<p>This is the password associated with the account specified by User Name. You specify this password in the default user account for the audit database and again in either the RDBMS profile or the ODBC data source definition in the ODBC.ini file on each NetPoint server connected to the audit database.</p> <p>If you specify a password in both the ODBC data source definition and the RDBMS profile, you should remain aware that the former stores the password string on each NetPoint host in unencrypted form in ODBC.ini, which is a plain text file, while the later stores the string in encrypted form on just the NetPoint LDAP directory server.</p>
Database	<p>This is the name of the target data source, which, in the case of the audit-to-database feature, is one of the following:</p> <ul style="list-style-type: none">• The name of the database containing the NetPoint audit data• A Microsoft Access database (.mdb file) for the Oblix Crystal Repository containing .gif image files and SQL-compatible queries used by the Crystal Report templates preconfigured to present Oblix audit information
Server	<p>This the name of the machine on which the RDBMS server (SQL Server or MySQL) resides.</p>
Port	<p>This is the port on which the RDBMS server listens for incoming requests.</p>
Driver	<p>The fully qualified path to the ODBC driver libraries on the local machine.</p>
Description	<p>Details to help you identify the data source definition.</p>

About ODBC Drivers

An ODBC driver library is specific to the type of database server to which you are connecting and the platform on which the driver is installed.

Each ODBC driver provides libraries that facilitate connection to the audit database.

An ODBC driver must exist on each machine hosting a NetPoint server that connects to the audit database. When both an Access Server and a COREid Server reside on the same machine, only a single ODBC driver is required for that host.

About the Windows ODBC Driver

By default, Windows installs the ODBC driver for SQL Server in the `\Windows\System32` directory. It is accessible through the ODBC Data Source Administrator, which you launch by navigating to `Start > Programs > Administrative Tools > Data Sources (ODBC)`.

The About tab in the ODBC Data Source Administrator displays the driver version number. If, for any reason, the installed version is lower than 3.5, or the driver is damaged or missing, you can download a replacement from the following web site:

`www.Microsoft.com\odbc`

The self-installing file is named `odbc35in.exe`.

About the Unix ODBC Driver

The Unix driver for connecting to a MySQL audit database is MyODBC 3.51. You can obtain the installation package from the following Web site:

`http://dev.mysql.com/downloads/connector/odbc/3.51.html`

The file should be named:

`MyODBC-Version-System.Machine.tar.gz`

where *Version* is the version number of the driver (such as “3.51-07”), *System* is the operating system platform (such as “sun-solaris2.8”), and *Machine* is the specific hardware platform (such as “sparc”) of the machine on which you will install the driver.

You should install the driver files in `/user/lib` or `/user/local/lib`. For procedure details see “Task overview: To install the MyODBC driver (Unix only)” on page 429.

About RDBMS Profiles

An RDBMS profile facilitates database auditing by specifying a primary ODBC data source (the NetPoint audit database) to which all NetPoint servers send audit data. An RDBMS profile can also specify secondary database instances for use in the event of failover.

NetPoint stores RDBMS profiles on the NetPoint directory server, where they are accessed by all the NetPoint servers connected to that directory server. You can configure an RDBMS profile through the Access System Console or the COREid System console. See the procedure “To create an RDBMS profile” on page 433. After you restart all your NetPoint servers, a new RDBMS profile is visible to all the servers connected to that directory server.

The name of each RDBMS profile on a directory server must be unique.

Each RDBMS profile supports a single NetPoint feature. Generally, Reporting (static reports) and Auditing (dynamic reports) share a single RDBMS profile, because they are considered to belong to the same “feature.” The MIIS provisioning feature uses a separate RDBMS profile.

All the NetPoint servers that use a particular feature (such as Reporting/Auditing) must use the same RDBMS profile.

By contrast, LDAP database profiles are server- and operation-specific. They can be shared by NetPoint servers, but they do not need to be. Thus, two or more NetPoint servers can each use a different LDAP database profile, even though each LDAP database profile is set up for the same LDAP server and operation.

Each database instance within an RDBMS profile specifies the information necessary to create a connection between a NetPoint server and the audit database. This includes the DSN (Data Source Name) for the ODBC data source definition that is actually used to connect to the audit database. It also includes a copy of the attributes listed in Table 61, “Key Attributes in an ODBC Data-Source Definition,” on page 406.

Because the same DSN appears in the ODBC.ini file of every machine that hosts a NetPoint server connected to the audit database, the details associated with the DSN stored in the RDBMS profile server must match exactly the details associated with every instance of that DSN in the ODBC.ini files throughout your NetPoint system.

If, for any reason, the associated attributes fail to match, the values for USER and PASSWORD recorded in the RDBMS profile take precedence over the corresponding values stored in ODBC.ini. On the other hand, the values for DATABASE and other attributes stored in ODBC.ini take precedence over the corresponding values in the RDBMS profile. In any case, the values in one location are never overwritten by the values stored in the other location.

About the NetPoint Audit Database

The NetPoint audit database collects audit data from all the Access Servers and COREid Servers in your NetPoint system. NetPoint supports ODBC 3.0-compliant databases on two types of database servers: SQL Server, which runs on the Windows platform, and MySQL running on Unix machines.

NetPoint auditing does not support MySQL Server running on the Windows platform.

About the Oblix/Crystal Repository

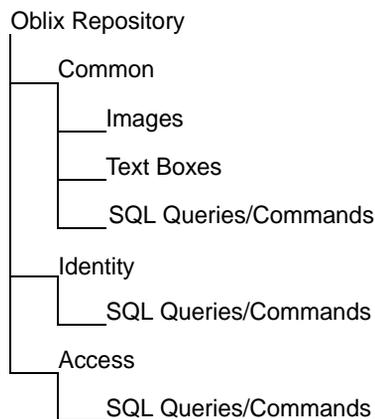
Within the context of the audit-to-database feature, the Oblix Repository and the Crystal Repository are synonymous because you link them through the orMap.ini file. See the procedure “To edit orMap.ini” on page 454.

This repository is a Microsoft Access format (.mdb) database that contains the following resources:

- .gif files used in NetPoint Audit Reports
- SQL queries and commands used in NetPoint Audit Reports
- Custom functions
- Templates that give NetPoint Audit Reports a consistent look and feel
- Sample reports

The following diagram reveals the organization of these resources within the repository.

Figure 15 Organization of the Oblix Repository



About NetPoint Audit Reports

Table 61, “Key Attributes in an ODBC Data-Source Definition,” on page 406 describes the NetPoint Audit Reports:

Table 62 Content types in the NetPoint Audit Reports

Audit Data Type	Audit Report Type	Description
Authentication Statistics	Authentication/Dynamic	The number of authentication success and authentication failures that occurred on a given server (or across the entire NetPoint system) during a given interval.
Authorization Statistics	Authorization/Dynamic	The number of authorization successes and authorization failures that occurred on a given server (or across the entire NetPoint system) during a given interval.
Access Failures by User	Authorization/Activity/Dynamic	The number of authorization requests from a given user that failed during a given interval.
Access Failures by Resource	Authorization/Activity/Dynamic	The number of authorization requests for a given resource that failed during a given interval.
Access Privileges	Filtered Query/Static	<p>Two types of access privilege reports are supported:</p> <ul style="list-style-type: none"> • All the users allowed to access a list containing one or more resources. • All the resources accessible by a list containing one or more users. <p>When this information is recorded to a file or database, it is referred to as a User Access Privilege Report or advanced Filtered Profile Query. See the procedure “To create and manage User access privilege reports” on page 449 for details.</p> <p>When simpler queries are displayed through the GUI, they are referred to as Access Tester output.</p> <p>This type of audit information is considered static, because it is derived from policy information that is stored on the directory server, rather than collected on a historical, event-by-event basis from either an Access Server or COREid Server.</p>
User Profile History	Identity Management/Dynamic	Changes to password, policy, and profile, etc. for all users.

Table 62 Content types in the NetPoint Audit Reports

Audit Data Type	Audit Report Type	Description
Group History	Identity Management/ Dynamic	A list of groups a given user has been added to or removed from within a given interval.
Revoked Users	Identity Management/ Dynamic	A list of users who have been locked out of the system.
Deactivated Users	Identity Management/ Dynamic	A list of users whose access accounts have been deactivated. (Lists of reactivated users can also be generated.)
Password Changes	Identity Management/ Dynamic	The number of passwords that have been changed throughout the system during a given interval.
User Status Changes	Identity Management/ Dynamic	The groups to which a given user or users has been added within a given interval.
Identity History	Identity Management/ Dynamic	Changes to password, policy, profile, and so on for one or more individual users.
Workflow Execution Time	Identity Management/ Dynamic	The average and maximum length of time it has taken to complete a workflow during a given period.

Setting Up File-Based Auditing

You turn file-based auditing on and off as well as change the name and location of the audit file generated by an individual Access Server or COREid Server through the NetPoint GUI. By default, the audit flag is off for all NetPoint Servers. The following two procedures detail the steps for activating and configuring file-based auditing for COREid Servers and Access Servers, respectively.

Note that you must activate the auditing flag for each NetPoint server individually.

If you wish, you may also modify the defaults for the following three categories of audit settings:

- **Common**—(See “To specify global COREid system events and profile attributes for audit” on page 441.)

- **NetPoint server-specific**—(See the procedures “To enable and configure auditing for each COREid Server” on page 439 and “To enable and configure auditing for each Access Server” on page 447.)
- **COREid application-specific**—(See the procedures “To specify User Manager events for audit” on page 442, “To specify Group Manager events for audit” on page 443, and “To specify Organization Manager events for audit” on page 444.)

The categories in the preceding list apply to both file-based and database auditing.

To configure file-based auditing for a COREid Server

1. Navigate to COREid System Console > System Admin > System Configuration > Configure COREid Server.
2. From the list, select the name of the COREid Server you wish to modify.
3. After the Details for COREid Server page appears, review the “audit to file” settings. If you wish to modify any of them, click the Modify button at the bottom of the page.

- After the Modify COREid Server page appears, change any of the Audit File parameters, as illustrated in the following screen shot:

Modify COREid Server

Name	np70-COREid1-7010									
Hostname*	<input type="text" value="brass"/>									
Port*	<input type="text" value="7010"/>									
Debug*	<input checked="" type="radio"/> Off <input type="radio"/> On									
Debug File Name*	<input type="text" value="/oblix/logs/debugfile.lst"/>									
Transport Security*	<input type="radio"/> Open <input checked="" type="radio"/> Simple <input type="radio"/> Cert									
Maximum Session Time (hours)*	<input type="text" value="24"/>									
Number of Threads*	<input type="text" value="20"/>									
Audit to Database Flag (auditing on/off)	<input checked="" type="radio"/> Off <input type="radio"/> On									
Audit to File Flag (auditing on/off)	<input checked="" type="radio"/> Off <input type="radio"/> On									
Audit File Name	<input type="text"/>									
Audit File Maximum Size (bytes)	<input type="text" value="100000"/>									
Audit File Rotation Interval (seconds)	<input type="text" value="7200"/>									
Audit Buffer Maximum Size (bytes)	<input type="text" value="25000"/>									
Audit Buffer Flush Interval (seconds)	<input type="text" value="7200"/>									
Log Threshold	Warning and above <input type="button" value="v"/>									
Log Handler Definitions	<table border="1"> <thead> <tr> <th>Name</th> <th>Log Level</th> <th>Output To</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> LogFatal2Sys</td> <td>Fatal</td> <td>System Log</td> </tr> <tr> <td><input type="checkbox"/> LogAll2File</td> <td>All Log Levels</td> <td>File</td> </tr> </tbody> </table>	Name	Log Level	Output To	<input type="checkbox"/> LogFatal2Sys	Fatal	System Log	<input type="checkbox"/> LogAll2File	All Log Levels	File
Name	Log Level	Output To								
<input type="checkbox"/> LogFatal2Sys	Fatal	System Log								
<input type="checkbox"/> LogAll2File	All Log Levels	File								
	<input type="button" value="Add"/> <input type="button" value="Delete"/>									
Scope File Name*	<input type="text" value="/oblix/logs/scopefile.lst"/>									
SNMP State*	<input checked="" type="radio"/> Off <input type="radio"/> On									
SNMP Agent Registration Port*	<input type="text"/>									

Note: If you change the fields marked with an asterisk(*), you must restart this COREid Server.

The following table describes the audit-to-file configuration parameters.

Table 63 Audit File Configuration Parameters

Parameter	Description	Default
Audit to File Flag	The radio buttons turn the audit to file feature to On or Off.	Off.
Audit File Name	<p>You can specify the absolute path and name of the audit file for the Access or COREid Server you are auditing.</p> <p>You may find it convenient to specify something similar to the following:</p> <p style="text-align: center;"><i>Component_install_dir</i>\oblix\log\auditfile.lst</p> <p>where <i>Component_install_dir</i> is the root installation directory for the associated Access or COREid Server.</p>	[blank]

Table 63 Audit File Configuration Parameters

Parameter	Description	Default
Audit File Maximum Size	The approximate size, in bytes, at which the existing audit file is closed and renamed to the following <i>AuditFileName.lst TimeStamp</i> where <i>AuditFileName</i> is the name of the audit file, and <i>TimeStamp</i> is a numerical representation, in seconds since midnight, January 1, 1971, of the moment when the file was created. By default, <i>AuditFileName</i> is <i>AuditFile</i> . Simultaneously, a new audit file named <i>AuditFileName</i> is created and opened for input.	100000
Audit File Rotation Interval	How often, in seconds, the audit file is renamed and a new one created to replace it. (See the table cell immediately above for details.) Time-based rotation occurs regardless of the current size of the audit file.	7200
Audit Buffer Maximum Size	The amount of audit data, in bytes, that can be accumulated in a buffer before the entire buffer is written to disk.	[blank]
Audit Buffer Flush Interval	The number of seconds after which the content of the audit buffer is written to the audit file regardless of the amount of data in the buffer.	7200

To configure file-based auditing for an Access Server

1. From the Access System Console, navigate to Access System Configuration > Access Server Configuration.
2. From the list in the right page, select the Access Server you want to modify.
3. After the “Details for Access Server” page appears, examine the audit file settings. If you wish to change any of them, click the “Modify” button at the bottom of the page.

- After the “Modify Access Server” page appears, change any of the Audit File parameters, as described in the preceding table and illustrated in the following screen shot.

Modify Access Server

Name	aaa-brass-7020
Hostname*	<input type="text" value="brass"/>
Port*	<input type="text" value="7020"/>
Debug*	<input checked="" type="radio"/> Off <input type="radio"/> On
Debug File Name*	<input type="text"/>
Transport Security*	<input type="radio"/> Open <input checked="" type="radio"/> Simple <input type="radio"/> Cert
Maximum Client Session Time (hours)*	<input type="text" value="24"/>
Number of Threads*	<input type="text" value="60"/>
Access Management Service*	<input type="radio"/> Off <input checked="" type="radio"/> On
Audit to Database (on/off)*	<input checked="" type="radio"/> Off <input type="radio"/> On
Audit to File (on/off)*	<input checked="" type="radio"/> Off <input type="radio"/> On
Audit File Name	<input type="text"/>
Audit File Size (bytes)	<input type="text" value="0"/>
Buffer Size (bytes)	<input type="text" value="512000"/>
File Rotation Interval (seconds)	<input type="text" value="0"/>
Engine Config Refresh Period (seconds)	<input type="text" value="14400"/>
URL Prefix Reload Period (seconds)	<input type="text" value="7200"/>
Password Policy Reload Period (seconds)	<input type="text" value="7200"/>
Maximum Elements in User Cache*	<input type="text" value="100000"/>
User Cache Timeout (seconds)*	<input type="text" value="1800"/>
Maximum Elements in Policy Cache *	<input type="text" value="10000"/>
Policy Cache Timeout (seconds)*	<input type="text" value="7200"/>
SNMP State*	<input checked="" type="radio"/> Off <input type="radio"/> On
SNMP Agent Registration Port*	<input type="text"/>

Please note that if you change the fields marked with an asterix(*), you will have to restart this Access Server.

Update Cache

Setting Up Database Auditing

To enable the NetPoint audit-to-database feature, you must complete the following sequence of high-level tasks, each of which consists of one or more procedures:

Task overview: Enabling database auditing

- Setup and verify your NetPoint environment.

For details, see “Setting Up Your NetPoint System for Database Auditing” on page 417.

2. Install and configure your RDBMS application (SQL Server or MySQL), then create and configure the NetPoint audit database.

For details, see “Setting up the Audit Database” on page 417.

3. Configure NetPoint for database auditing.

This involves enabling your NetPoint servers to connect to the audit database by creating ODBC data source definitions and an RDBMS profile. You also need to configure and verify both your COREid and Access systems for auditing. For details, see “Configuring NetPoint Auditing” on page 437.

4. Install and configure Crystal Reports, then verify that the NetPoint audit templates can display audit database information.

For details, see “Setting up NetPoint Audit Reports” on page 451.

Setting Up Your NetPoint System for Database Auditing

Before you can use the audit-to-database feature, you must verify that all the Access Server and COREid Server hosts in your NetPoint system are running Windows, or that all are running Unix. In other words, the audit-to-database feature is not supported for NetPoint domains in which some servers connected to the audit database run on Windows hosts while others run on Unix machines.

This prohibition against mixing platforms within a NetPoint environment applies to only the machines hosting NetPoint servers that connect to the ODBC database and to the machine hosting the ODBC database server; the machine(s) hosting the NetPoint LDAP server can run either Windows or Unix. The machine hosting Crystal Reports must run Windows, regardless of the type of ODBC database being used. NetPoint servers not connected to the audit database can run on any platform.

Setting up the Audit Database

The NetPoint audit database is an ODBC 3.0 compliant database running on SQL Server or MySQL for Unix.

Task overview: Setting up NetPoint for the audit database

1. Install SQL Server if your NetPoint servers are running Windows.

See “About installing SQL Server (Windows)” on page 418. Alternatively, install MySQL if all your NetPoint servers are running Unix. (See “About installing MySQL (Unix)” on page 419.)

2. Create and configure the NetPoint audit database.

See the procedures “To create the audit database (SQL Server/Windows)” on page 423 or “To create the audit database (MySQL/Unix)” on page 423.

3. Upload the NetPoint auditing and reporting schema to the auditing database.
See “Task overview: Uploading the audit schema” on page 424.
4. Create an ODBC data source definition (System DSN) on each NetPoint server that will send data to the audit database.
See the procedures “To create an ODBC data source definition (Windows)” on page 431 or “To modify the “myodbc3” data source definition in the ODBC.ini file (Unix only)” on page 430.
5. Create an RDBMS profile on the NetPoint LDAP directory server so that each NetPoint server connected to the NetPoint directory server can recognize the ODBC data source definition on its host machine.
See the procedure “To create an RDBMS profile” on page 433.
6. Make the RDBMS profile “visible” system-wide by restarting all your NetPoint servers.
See the procedures “To make the RDBMS profile visible (Windows)” on page 436 or “To make the RDBMS profile visible (Unix)” on page 437.

Installing the ODBC Database Server

You install SQL Server if all the NetPoint server hosts in your system run Windows. You install MySQL if all your NetPoint server hosts run Unix.

About installing SQL Server (Windows)

You can use the Standard, Enterprise, or Developer Edition of SQL Server 2000.

If you plan to implement other NetPoint features that use SQL Server (for example, SharePoint Portal Server integration or MIIS provisioning integration), the auditing feature can share a single SQL Server installation with these other features, provided that SQL installation meets the minimum requirements dictated by each feature.

Follow the instructions supplied by Microsoft to install SQL Server. The installation wizard prompts you to specify setup options. In most cases, you should accept the defaults as you progress through the wizard pages, but first check the following table and enter any settings that differ from the defaults:

Table 64 Special Settings for SQL Server Installation

Wizard Page Setting	What to Specify
autorun.exe opening screen	SQL Server 2000 Components > Install Database Server
Installation target	“Local Computer”

Table 64 Special Settings for SQL Server Installation

Wizard Page Setting	What to Specify
Installation option	“Create a new instance of SQL Server”
Type of installation	“Server and Client Tools”
Instance name	“Default”
Type of setup	“Typical”
Services accounts	“Use the same account for each service. Auto Start SQL User Service”
Service settings	“Use Local System account” The default login name, which is also referred to as the Login ID or User Name, is “sa,” and the password can be blank if the box labeled “blank password” is checked. The password can be whatever you wish if “blank password” is not checked. In any case, record the login name and associated password so that you can duplicate them exactly when you create your RDBMS profile and the ODBC data source definitions on each NetPoint server host.
Authentication mode	“Mixed Mode”

After you have successfully installed SQL Server, proceed to the procedure “To create the audit database (SQL Server/Windows)” on page 423.

About installing MySQL (Unix)

To install MySQL and configure both MySQL and your host environment so they can support NetPoint database auditing, you must complete the sequence of procedures detailed in the following sections.

Task overview: To install and configure MySQL

1. Set up the MySQL user account.
See “To set up the MySQL user account” on page 420.
2. Obtain and unpack the appropriate MySQL installation package.
See “To obtain the MySQL installation package” on page 420.
3. Make the MySQL installation “visible” by creating a symbolic link and setting the PATH environment variable.
See “To make the MySQL installation visible” on page 420.

4. Run the MySQL installation script.
See “To run the MySQL installation script” on page 421.
5. Configure MySQL for transactional support.
See “To configure MySQL for transactional support” on page 421.
6. Enable remote connections to MySQL by setting the default user password, then granting full privileges to the default user and assigning ownership of the MySQL directories.
See “To enable remote connections to MySQL” on page 422.

To set up the MySQL user account

1. From the system prompt of the machine that will host MySQL server, enter the following command to make yourself the root user:

```
shell> su root
```

2. Enter the following command to create a group named “mysql.”

```
shell> groupadd mysql
```

3. Enter the following command to create a user named “mysql” in the group named “mysql.”

```
shell> useradd -g mysql mysql
```

4. Proceed to “To obtain the MySQL installation package” on page 420

To obtain the MySQL installation package

1. From the machine that will host MySQL, point your web browser to the following web site:

```
http://dev.MySQL.com/
```

2. Follow the prompts to download the installation package of the version of MySQL appropriate for your machine to a temporary folder on your hard drive.

The NetPoint audit-to-database feature has been tested against MySQL, Standard Edition, version 4.0 for Sparc/Solaris 2.8, which is packaged in the following file:

```
mysql-standard-4.0.20-pc-solaris2.8-i386.tar.gz
```

3. Decompress and extract the contents of the installation package to the installation directory of your choice.
4. Proceed to “To make the MySQL installation visible” on page 420

To make the MySQL installation visible

1. Enter the following command to change to the local user directory.

```
shell> cd usr/local
```

2. Enter the following command to create a symbolic link to MySQL.

```
shell> ln -s MySQL_dir/MySQL_version mysql
```

where *MySQL_dir* is the full path to the directory in which you install MySQL, and *MySQL_version* is the full version name of the MySQL package you are installing. For example, the NetPoint audit-to-database feature was tested against the following version of MySQL:

```
mysql-standard-4.0.20-pc-solaris2.8-i386
```

3. Enter the following command to add the MySQL directory to the existing PATH environment variable so that the MySQL libraries are visible.

```
shell> setenv PATH MySQL_dir: {$PATH}
```

where *MySQL_dir* is the directory in which you install MySQL.

4. Proceed to “To run the MySQL installation script” on page 421.

To run the MySQL installation script

1. Enter the following command to change the MySQL directory.

```
shell> cd MySQL_dir
```

where *MySQL_dir* is the directory in which you install MySQL.

2. Enter the following command to run the script “mysql_db” that initializes the MySQL grant tables with the default user set to “mysql.”

```
shell> scripts/mysql_db --user=mysql
```

3. Proceed to “To configure MySQL for transactional support” on page 421.

To configure MySQL for transactional support

1. Enter the following command to install the “Berkeley database” option, which enables transactional support for MySQL.

```
shell> ./configure ---with-berkeley-db
```

This command also starts the MySQL database server.

If at some point in the future, you need to start MySQL without the Berkeley option, use the following command.

```
shell> bin/myslqd_safe --user=mysql &
```

Oblix recommends that you use the following pair of commands to start and stop MySQL under normal conditions.

```
shell> .support-files/mysql.server start
```

```
shell> .support-files/mysql.server stop
```

2. Proceed to “To enable remote connections to MySQL” on page 422.

To enable remote connections to MySQL

1. Enter the following commands to grant the root and default users (root and mysql, respectively) system permission to modify key directories:

```
shell> cd MySQL_dir
shell> chown -R root      .
shell> chown -R mysql    data
shell> chgrp -R mysql    .
```

where *MySQL_dir* is the MySQL installation directory.

The second command grants ownership of the binaries in the MySQL installation directory and all subdirectories to root. The third command grants ownership of the data directory and all its subdirectories to the default database user, mysql. The fourth and final command changes the group.

2. Enter the following command to ensure that MySQL recognizes the password you want for the default account.

```
shell> .bin/mysqladmin -u root password pwd
```

where *pwd* is the password you prefer.

3. Enter the following command to confirm that MySQL recognizes your preferred password.

```
shell> bin/mysql --user=root mysql -p
```

(Enter the password for the MySQL default user at the prompt.)

4. Enter the following SQL commands at the MySQL prompt to grant all privileges to the default user (mysql).

```
mysql> INSERT INTO user (Host, User, Password) VALUES ('%',
  'mysql', PASSWORD, ('pwd'));
```

where *pwd* is the password you set for the default user mysql. (do not forget to place a semicolon at the end of every line containing an SQL statement.)

```
mysql> INSERT INTO user (Host, User, Password) VALUES
  ('localhost', 'mysql', PASSWORD, ('pwd'));
```

where *pwd* is the password you set for the default user mysql.

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP,
  ALTER ON db_name * TO mysql IDENTIFIED BY 'pwd';
```

where *db_name* identifies your audit database, and *pwd* is the password of your default user, mysql.

Alternatively, you can achieve the same result by entering the following command:

```
mysql> GRANT ALL PRIVILEGES on *.* TO 'mysql' @ 'host_name';
```

where *host_name* identifies the machine hosting MySQL.

5. Enter the following command to register the changes in the audit database.

```
mysql> FLUSH PRIVILEGES;
```

6. Enter the following command to log off MySQL.

```
mysql> quit;
```

7. Proceed to “To create the audit database (MySQL/Unix)” on page 423.

Creating the NetPoint Audit Database

The procedure for creating the NetPoint audit database differs depending on whether you are using SQL Server or MySQL.

To create the audit database (SQL Server/Windows)

1. On the machine hosting SQL Server, navigate to:

```
My Computer > Manage > Services and Applications > Microsoft SQL  
Servers > hostname
```

where *hostname* is the Windows Services name for the machine hosting SQL Server.

2. In the left pane of the Computer Management window, right-click Databases in the branch beneath the host name of the machine on which SQL Server is installed, then click New Database.
3. Select a descriptive name for the database, then click OK.

For instance, NPAuditDB stands for “NetPoint audit database.” An icon representing the new database appears in the right hand pane of the Computer Management window.

4. Proceed to “Uploading the NetPoint Audit Schema” on page 424.

To create the audit database (MySQL/Unix)

1. Enter the following command to log on to MySQL Server as the default user, “mysql.”

```
shell> bin/mysql --user=mysql
```

2. Enter the following command to create the audit database.

```
mysql> create database db_name;
```

where *db_name* is the name of the NetPoint audit database you are creating.

Alternatively, you can simply upload the NetPoint audit schema to an existing database.

3. Proceed to “Uploading the NetPoint Audit Schema” on page 424.

Uploading the NetPoint Audit Schema

The NetPoint audit schema are objects that allow you to import audit data from the NetPoint servers and export that data to the Oblix/Crystal Repository, where it is presented in NetPoint audit reports.

Task overview: Uploading the audit schema

1. Copy the NetPoint audit schema and supporting resources from a NetPoint server host to the NetPoint audit database host.

The copy procedure differs depending whether you are performing a Windows-to-Windows or a Unix-to-Unix transfer. See the procedure “To copy the audit schema to the audit database host:” on page 424.

2. Upload the audit schema to your audit database.

This upload procedure differs depending on whether you are using SQL Server or MySQL. See the procedure “To upload the audit schema (SQL Server/Windows)” on page 425 or “To upload and verify the audit schema (MySQL/Unix)” on page 426.

3. Verify that the schema have been uploaded successfully.

This procedure differs depending on whether you are using SQL Server or MySQL. See the procedure “To verify the audit schema in (SQL Server/Windows)” on page 427 or “To upload and verify the audit schema (MySQL/Unix)” on page 426.

To copy the audit schema to the audit database host:

1. On any machine hosting a NetPoint server, locate the directory containing the NetPoint audit schema by navigating to:

```
Component_Install_dir\oblix\reports
```

where *Component_Install_dir* is the root installation directory of your NetPoint server.

2. Using any of the means available for your particular operating system and network domain, copy the file `audit.sql` to a directory on the machine hosting your NetPoint auditing database. (This procedure isn't necessary if you happened to install your audit database on the same machine as one of your NetPoint servers.) If you are using MySQL, you must also copy `audit_mysql_ext.sql` to the same target directory as `audit.sql`.
3. As appropriate for the database application you are using, proceed to one of the following:
 - “To upload the audit schema (SQL Server/Windows)” on page 425
 - “To upload and verify the audit schema (MySQL/Unix)” on page 426

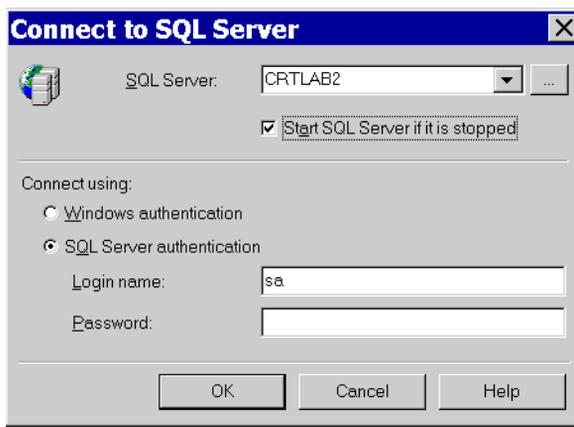
To upload the audit schema (SQL Server/Windows)

1. On the machine hosting SQL Server, navigate to:

Start > Programs > Microsoft SQL Server > Query Analyzer

2. If the “Connect to SQL Server” page is not already displayed in the SQL Query Analyzer window, navigate to:

File > Connect



3. In the Connect to SQL Server page, verify that the Windows Service name of your SQL Server host is displayed in the field labeled SQL Server.
4. Check “Start SQL Server if it is stopped.”
5. Set “Connect using” to “SQL Server authentication.”
6. Enter whatever login name and password you selected when installing SQL Server, then click OK to commit your choices. A Query window will open within the SQL Query Analyzer window.
7. Launch the NetPoint audit database in the SQL Query Analyzer. In the SQL Query Analyzer menu, navigate to:

File > Open

8. Navigate to “audit.sql” which is located under the directory you copied from your NetPoint server to your audit database host in the preceding procedure.

For details, see the procedure “To copy the audit schema to the audit database host:” on page 424. The specific location of audit.sql is:

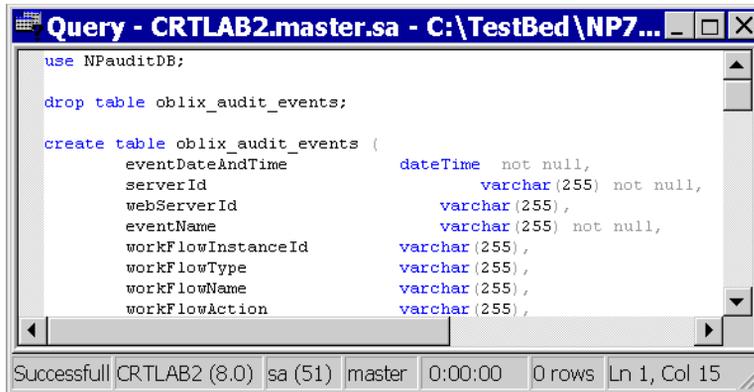
```
..\reports\crystal\audit.sql
```

9. In the Query window, add the following line to the very beginning of the file audit.sql:

```
use AuditDBName;
```

where *AuditDBName* specifies the NetPoint audit database you created in the procedure “To create the audit database (SQL Server/Windows)” on page 423. In our example, we named the database NPAuditDB.

For all SQL statements, don't forget to place a semi-colon at the end of the line.



```
Query - CRTLAB2.master.sa - C:\TestBed\NP7...
use NPAuditDB;

drop table oblix_audit_events;

create table oblix_audit_events (
    eventDateAndTime    dateTime not null,
    serverId             varchar (255) not null,
    webServerId         varchar (255) ,
    eventName           varchar (255) not null,
    workFlowInstanceId  varchar (255) ,
    workFlowType        varchar (255) ,
    workFlowName        varchar (255) ,
    workFlowAction      varchar (255) ,
)

Successfull CRTLAB2 (8.0) sa (51) master 0:00:00 0 rows Ln 1, Col 15
```

10. Press F5 to execute the command. Alternatively, select Query > Execute from the SQL Query Analyzer menu.

The first time you do this, the application will return the following error message:

```
cannot drop the table 'oblix_audit_reports', because it does
not exist in the system catalog yet
```

This is both customary and logical, because the table did not exist when the “use” command was executed. If you save audit.sql and subsequently re-execute this command, the error message will not reappear, because the table now exists.

Minimize, but do not close the Query window; you will need to add another line to audit.sql when you verify that the schema have uploaded successfully. Proceed to: “To verify the audit schema in (SQL Server/Windows)” on page 427.

To upload and verify the audit schema (MySQL/Unix)

1. Open the files audit.sql and audit_mysql_ext.sql in any plain text editor.
2. Cut-and-paste the entire contents of both files into the MySQL window that opened when you created the audit database. You only have to do this once for the two files; the schema are now entered permanently in the audit database.

3. Enter the following command to verify that the schema have been uploaded successfully.

```
mysql> SHOW TABLES;
```

The four tables you just created will display.

4. Proceed to “Enabling NetPoint Servers to Connect to the Audit Database” on page 428.

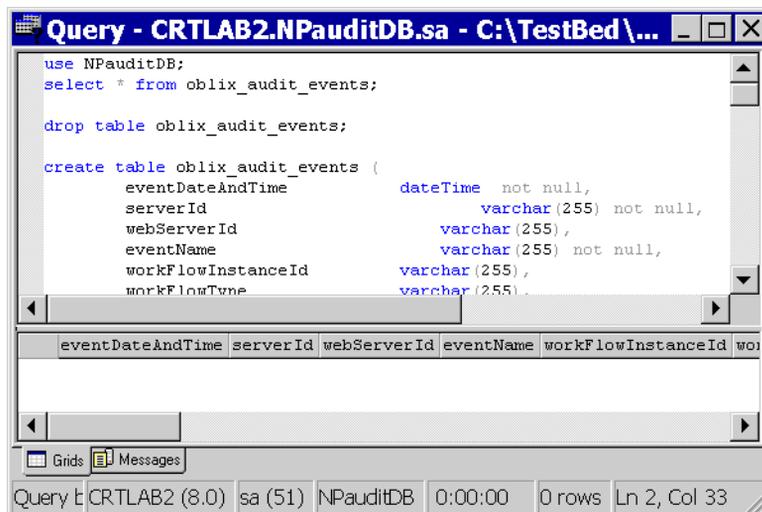
To verify the audit schema in (SQL Server/Windows)

1. Perform a “dummy” select from the oblix_audit_events table. Add the following line immediately beneath the line you added to audit.sql in the procedure “To upload the audit schema (SQL Server/Windows)” on page 425”:

```
select * from oblix_audit_events;
```

do not forget to include a semi-colon at the end of the line.

2. Click F5 to execute the command.



Column headings such as “eventDateAndTime” will appear in a pane immediately beneath the code pane in the Query window. These indicate that the audit.sql schema uploaded successfully.

3. In the SQL Query Analyzer window, click File > Save to record the changes to your audit.sql, which is now linked to your NetPoint audit database. Proceed to “Enabling NetPoint Servers to Connect to the Audit Database” on page 428.

Enabling NetPoint Servers to Connect to the Audit Database

You enable your NetPoint servers to connect to the audit database by creating a RDBMS profile on the NetPoint directory server and ODBC data source definitions on each machine hosting a NetPoint server that connects to the audit database. A single, unique System DSN (System-wide Data Source Name) connects all of these objects.

It is extremely important that every attribute associated with a given DSN in both the RDBMS profile and the ODBC data source definitions on the NetPoint server hosts match exactly. For details, see “To create a primary RDBMS instance” on page 435.

Task overview: Enabling NetPoint servers to connect to the audit database

1. **Unix**—install the MyODBC driver on each NetPoint server host and set the environment variables and other links necessary to make the driver “visible.” See: “Task overview: To install the MyODBC driver (Unix only)” on page 429 and “To make the MyODBC driver visible” on page 429.
2. **Unix**—enable a connector to link NetPoint to the audit database by editing the ODBC.ini file on each NetPoint server host. see the procedure “To modify the “myodbc3” data source definition in the ODBC.ini file (Unix only)” on page 430.
3. **Windows**—create an ODBC data source definition (System DSN) on each NetPoint Server host. See the procedure “To modify the “myodbc3” data source definition in the ODBC.ini file (Unix only)” on page 430.
4. **All**—Using either the COREid System Console or the Access System Console, create an RDBMS profile on the directory server. See: “Task overview: To set up an RDBMS profile” on page 433. This includes the following tasks:
 - a) Create a primary RDBMS instance. See the procedure “To create a primary RDBMS instance” on page 435.
 - b) Create optional secondary RDBMS instances for your RDBMS profile. See: “Task overview: To create a secondary RDBMS instance” on page 436
 - c) Restart all NetPoint servers so that the RDBMS profile is “visible” system-wide. See the procedure “To make the RDBMS profile visible (Windows)” on page 436.

Task overview: To install the MyODBC driver (Unix only)

1. Obtain and unpack the installation package for the version of the MyODBC database driver appropriate for your installation. (See “To obtain and unpack the installation package for MyODBC” on page 429.)
2. Make the MyODBC driver visible by creating a symbolic link and setting an environment variable. (See “To make the MyODBC driver visible” on page 429.)

To obtain and unpack the installation package for MyODBC

1. From a machine that hosts a NetPoint server you want to connect to the audit database, point your web browser to the following web site.

```
http://dev.mysql.com/downloads/connector/odbc/3.51.html
```

2. Follow the prompts to download the installation package of the MySQL version appropriate for your machine to a temporary folder on your hard drive.

The NetPoint audit-to-database feature has been tested against MyODBC, version 3.51 for Sparc/Solaris 2.8, which is packaged in the following file:

```
MyODBC-3.51.06-sun-solaris2.8-sparc.tar.gz
```

3. Decompress and extract the contents of the installation package to the directory “/user/lib” or “/usr/local/lib” or any other path you prefer.
4. Repeat this procedure on every NetPoint server you plan to connect to the audit database.
5. Proceed to “To make the MyODBC driver visible” on page 429.

To make the MyODBC driver visible

1. Enter the following command to create a symbolic link for the MyODBC driver.

```
shell> ln -s MyODBC_dir/MyODBC_version myodbc
```

where *MyODBC_dir* is the full path to the installation directory for MyODBC and *MyODBC_version* is the full version name of the MyODBC package you have installed. For example, the NetPoint audit-to-database feature was tested against the following version of MyODBC:

```
MyODBC-3.51.06-sun-solaris2.8-sparc.tar.gz
```

2. Enter the following commands to set the environment variables that will make the MyODBC installation visible.

```
shell> export ODBCINI=/usr/local/etc/odbc.ini
```

```
shell> export ODBSYSCINI=/usr/local/etc
```

3. Repeat this procedure on every NetPoint server you plan to connect to the audit database.

4. Proceed to “To modify the “myodbc3” data source definition in the ODBC.ini file (Unix only)” on page 430.

To modify the “myodbc3” data source definition in the ODBC.ini file (Unix only)

1. On any NetPoint Server where you have installed the MyODBC driver, move the file ODBC.ini from the MyODBC installation directory to the following directory

/usr/local/etc

2. Use any plain text editor to open ODBC.ini for edit.
3. Verify that the settings in the “myodbc3” section (which is the default ODBC data source definition for the NetPoint audit-to-database feature on Unix platforms) conform to the values in the following table.

Table 65 Parameters for the myODBC3 Section of ODBC.ini

Parameter	Required Value	Description
Driver	/usr/local/lib/libmyodbc3.so	The fully qualified path to the MyODBC driver.
Description	[user preference]	A short note to identify this data source definition.
SERVER	localhost	The name of the machine on which the audit database resides.
PORT	[user preference]	The port MySQL listens to for incoming requests.
USER	mysql	The default user, MySQL.
Password	[leave this field empty]	The password for the MySQL default user, MySQL. Leave this empty. The RDBMS and database values for this parameter will be used.
Database	auditdb	The name of the audit database.
Option	3	The ODBC compatibility level. Do not change the default, which is 3.
Socket	[leave untouched]	Not used.
TraceFile	[optional]	The fully qualified path and name of the trace file.
Trace	[required only if you specify TraceFile]	“1” turns the trace log on; “0” turns trace log off.

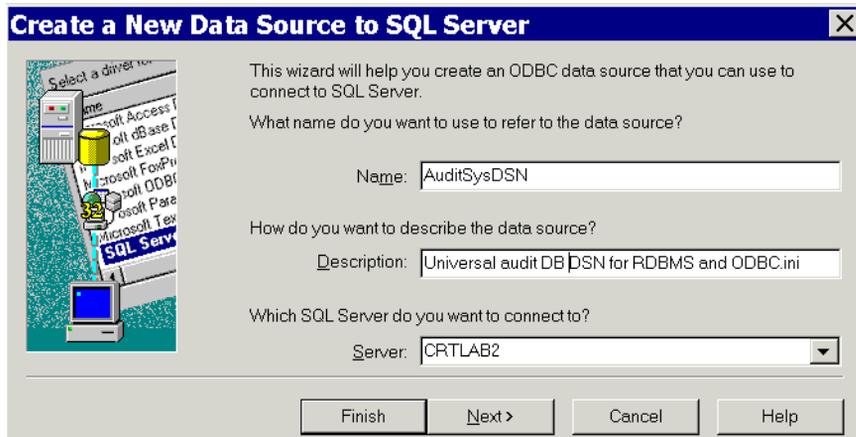
4. Save odbc.ini.

5. Repeat this procedure on each NetPoint server host you wish to connect to the audit database.
6. Proceed to “Task overview: To set up an RDBMS profile” on page 433.

To create an ODBC data source definition (Windows)

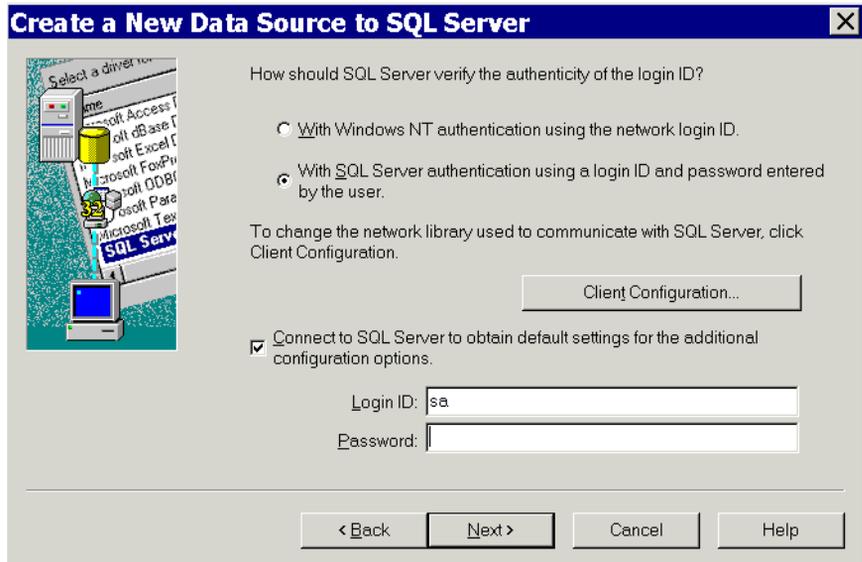
1. On a NetPoint server host you wish to connect to the audit database, navigate to: Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC).
2. Click the System DSN tab.
3. Click Add.
4. From the drop-down list of database drivers, select SQL Server, then click Finish.
5. In the Name field, type a descriptive name.

For instance, AuditSysDSN stands for the System DSN for the audit database. Write this name down, because you will have to use this exact character string for the ODBC data source definitions on every other NetPoint Server host, and for the primary RDBMS instance in your RDBMS profile as well.

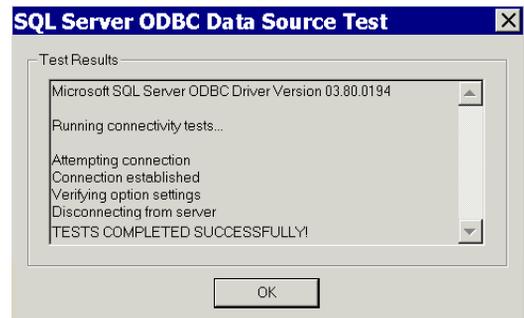
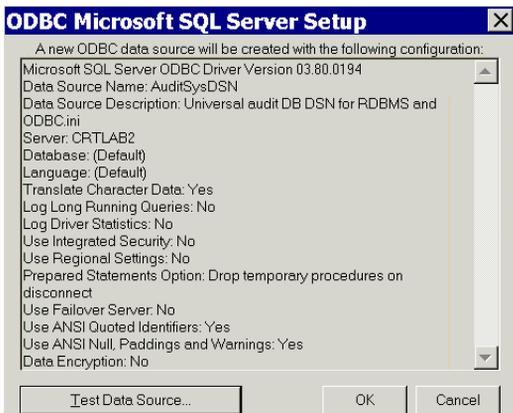


6. In the Description field enter notes to help users identify this object.
7. In the Server field, select the Windows Services name of the host on which the NetPoint audit database is running, then click Next.

8. When the next page appears, select “With SQL server authentication. . .”



9. Verify that “Connect to SQL server to obtain . . .” is selected.
10. Type the Login ID and password you specified when you installed SQL Server.
11. Leaving the default settings on the next two pages untouched, click Next, then click Finish.
12. After a page appears listing the settings for the new ODBC data source definition, click Test Data Source.



13. After a page appears to announce success, click OK three times to dismiss the open pages.

14. Repeat this procedure on every NetPoint server host you wish to connect to the audit database.

Make sure you use the exact same settings in every case, and for your RDBMS database instances as well. Proceed to: “Task overview: To set up an RDBMS profile” on page 433.

Task overview: To set up an RDBMS profile

1. Create an RDBMS profile. See the procedure “To create an RDBMS profile” on page 433.
2. Create a primary RDBMS instance. See the procedure “To create a primary RDBMS instance” on page 435.
3. Create (optional) secondary RDBMS instances. See: “Task overview: To create a secondary RDBMS instance” on page 436.
4. Make the RDBMS profile visible. As appropriate for the database application you are using, See the procedure “To make the RDBMS profile visible (Windows)” on page 436 or “To make the RDBMS profile visible (Unix)” on page 437.

To create an RDBMS profile

1. From any COREid Server host, navigate to:

```
COREid System Console > System Admin > System Configuration >  
Configure Directory Options > Configure RDBMS Profiles > Add
```

Alternatively, you can create the RDBMS profile from any Access Server host by navigating to:

Access System Console > System Configuration > View Server Settings > Configure RDBMS Profiles > Add

The Create RDBMS Profile page is identical, whether you access it through the COREid System Console or the Access System Console.

Create RDBMS Profile

Name*

Used By Reporting Auditing MIIS

Database Instances

Name	Machine	Port number	Server Type
<input type="button" value="Add"/>			
Maximum Active Servers	<input type="text" value="1"/>		
Failover Threshold	<input type="text" value="1"/>		
Sleep For (Seconds)	<input type="text" value="60"/>		
Max. Session Time (Min.)	<input type="text" value="0"/>		

Enable Profile

Note: The fields marked with an asterisk(*) are required fields.

2. In the Name field, enter a descriptive name. For instance AuditDBSysDSN refers to the System DSN created for the audit database. Technically, we are creating an RDBMS profile here, but this name provides a convenient universal name to identify matching sets of data source definition values in the RDBMS profile and the ODBC.ini files on each NetPoint server host.
3. In the Used By field, check Reporting and Auditing.
4. Verify that the Enable Profile box is selected. Proceed to “To create a primary RDBMS instance” on page 435.

To create a primary RDBMS instance

1. In the Create RDBMS Profile page, click the Add button next to the table labeled Database Instances.

Create Database Instance

Name*	<input type="text" value="AuditDBSysDSN"/>
DSN Name*	<input type="text" value="AuditDBSysDSN"/>
Database Name	<input type="text" value="NPAuditDB"/>
User name	<input type="text" value="sa"/>
Password	<input type="password"/>
Time Limit	<input type="text" value="0"/>
Size Limit	<input type="text" value="0"/>
Initial Connections	<input type="text" value="5"/>
Maximum Connections	<input type="text" value="5"/>

Note: The fields marked with an asterisk(*) are required fields.

Changes made to this DB Instance require that you save the DB profile too.

2. In the Name field of the Create Database Instance page, enter a descriptive name. For convenience, you can use the universal name you gave to the RDBMS Profile, such as AuditDBSysDSN.
3. In the DSN name field, enter a descriptive name. For convenience, you can use the name you just gave to both the database instance and the RDBMS profile. In our example, we have been using AuditDBSysDSN. (For Unix systems, Oblix recommends that you use the DSN “myodbc3,” which appears in the copy of ODBC.ini that is installed as part of the MyODBC installation package.

Important: The character string you specify as the DSN for your RDBMS instance must match exactly the DSN you specify for the ODBC data source definition on each NetPoint server. Furthermore, the values for all other database instance attributes must be empty or match exactly the values for the corresponding attributes in the ODBC data source definitions throughout your NetPoint system.

4. In the Database field, specify the name of the NetPoint audit database. Our example uses NBAuditDB.
5. In the User name field, enter the login name you specified when you created the NetPoint audit database.
6. Enter the password associated with the audit database login name.

7. Leave the other fields at their default settings. You can change them later, if necessary. Click Save to commit the database instance settings you have entered.
8. When the Modify RDBMS Profile page appears, click Save to commit the RDBMS profile settings you have entered.
9. If you wish to create a secondary RDBMS instance, proceed to the task overview immediately following. Otherwise, proceed to “To make the RDBMS profile visible (Windows)” on page 436.

Task overview: To create a secondary RDBMS instance

1. Perform all the steps in “Creating the NetPoint Audit Database” on page 423. For convenience, you may want to name the second instance of the audit database something like NPAuditDB_2.
2. Perform all the steps in “Uploading the NetPoint Audit Schema” on page 424.
3. Perform steps 5 through 11 in “To create an RDBMS profile” on page 433. For convenience, you may want to specify the name of the RDBMS instance and the DSN name as something like AuditDBSysDSN_2.
4. After the Modify RDBMS Profile page appears, verify that the Server Type for your secondary RDBMS instance is set to secondary.
5. Add the ODBC data source definitions for the secondary RDBMS instance (s) to ODBC.ini on each NetPoint server host.

As appropriate for the database application you are using, see the procedure “To create an ODBC data source definition (Windows)” on page 431 or “To modify the “myodbc3” data source definition in the ODBC.ini file (Unix only)” on page 430.

6. As appropriate for the database application you are using, proceed to “To make the RDBMS profile visible (Windows)” on page 436 or “To make the RDBMS profile visible (Unix)” on page 437.

To make the RDBMS profile visible (Windows)

1. On any NetPoint server host, navigate to My Computer > Manage > Services and Applications > Services.
2. Right-click the icon representing the NetPoint server on the machine, then select Restart from the dropdown menu.

If you installed both an Access Server and a COREid Server on the same machine, perform this procedure for both servers.

3. Repeat this procedure for all the NetPoint server hosts you wish to connect to the audit database.
4. Proceed to “Configuring NetPoint Auditing” on page 437.

To make the RDBMS profile visible (Unix)

1. On a machine hosting a NetPoint server, run one of the following commands to stop your NetPoint server.
 - **Access Servers**—`stop_access_server`
 - **COREid Servers**—`stop_ois_server`
2. Run one of the following commands to start your NetPoint server.
 - **Access Servers**—`start_access_server`
 - **COREid Servers**—`start_ois_server`
3. Repeat this procedure for all the NetPoint server hosts you wish to connect to the audit database.
4. Proceed to: “Configuring NetPoint Auditing” on page 437.

Configuring NetPoint Auditing

You must configure NetPoint for both file-based and database auditing.

By default, both file-based auditing and database auditing are turned off for all NetPoint servers, so you must manually enable file-based and/or database auditing for each NetPoint server in your system.

You use the NetPoint graphical user interface to configure audit options on a system-wide, per server, per event, and per application basis. For a table of NetPoint audit options, the GUI locations from where they can be set, and the scope they cover, see “About NetPoint Audit Options” on page 400.

The NetPoint defaults for auditing options are optimal for most situations. However, you do need to turn on the type or types of auditing you want on the NetPoint servers you want to audit. If you send data to the audit database, you must also replace the default audit data format string on both the COREid and Access systems. See “To modify audit output formatting for the COREid system” on page 440 and “To modify audit output formatting for the Access system” on page 448. The following task includes both mandatory and optional configuration procedures for NetPoint auditing.

Task overview: To configure NetPoint auditing

1. Turn on file-based and/or database auditing for individual COREid Servers, and modify audit file attributes, if you wish.

See the procedure “To enable and configure auditing for each COREid Server” on page 439.
2. Globally modify the audit output formatting for the COREid system.

See the procedure “To modify audit output formatting for the COREid system” on page 440.

3. Specify what data for which COREid events will be audited. This includes the following categories:
 - a) Events common to the User, Group, and Organization Manager applications. See the procedure “To specify global COREid system events and profile attributes for audit” on page 441.
 - b) User manager events. See the procedure “To specify User Manager events for audit” on page 442.
 - c) Group Manager events. See the procedure “To specify Group Manager events for audit” on page 443.
 - d) Organization Manager events. See the procedure “To specify Organization Manager events for audit” on page 444.
4. Verify that all COREid Servers can record data to the audit database. See the procedure “To verify that all COREid Servers can record data to the audit database (Windows)” on page 444.
5. Turn on file-based and/or database auditing for individual Access Servers, and modify audit file attributes, if you wish. See the procedure “To enable and configure auditing for each Access Server” on page 447.
6. Globally modify the audit output formatting for the Access system. See the procedure “To modify audit output formatting for the Access system” on page 448.
7. Create and manage User access privilege reports. See the procedure “To create and manage User access privilege reports” on page 449.

To enable and configure auditing for each COREid Server

1. Navigate to COREid System Console > System Administration > System Configuration > Configure COREid Server > *ServerName* > Modify where *ServerName* specifies the COREid Server you want to modify.

Modify COREid Server

Name	np70-COREid1-7010									
Hostname*	<input type="text" value="brass"/>									
Port*	<input type="text" value="7010"/>									
Debug*	<input checked="" type="radio"/> Off <input type="radio"/> On									
Debug File Name*	<input type="text" value="/oblix/logs/debugfile.lst"/>									
Transport Security*	<input type="radio"/> Open <input checked="" type="radio"/> Simple <input type="radio"/> Cert									
Maximum Session Time (hours)*	<input type="text" value="24"/>									
Number of Threads*	<input type="text" value="20"/>									
Audit to Database Flag (auditing on/off)	<input checked="" type="radio"/> Off <input type="radio"/> On									
Audit to File Flag (auditing on/off)	<input checked="" type="radio"/> Off <input type="radio"/> On									
Audit File Name	<input type="text"/>									
Audit File Maximum Size (bytes)	<input type="text" value="100000"/>									
Audit File Rotation Interval (seconds)	<input type="text" value="7200"/>									
Audit Buffer Maximum Size (bytes)	<input type="text" value="25000"/>									
Audit Buffer Flush Interval (seconds)	<input type="text" value="7200"/>									
Log Threshold	<input type="text" value="Warning and above"/>									
Log Handler Definitions	<table border="1"> <thead> <tr> <th>Name</th> <th>Log Level</th> <th>Output To</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> LogFatal2Sys</td> <td>Fatal</td> <td>System Log</td> </tr> <tr> <td><input type="checkbox"/> LogAll2File</td> <td>All Log Levels</td> <td>File</td> </tr> </tbody> </table> <p><input type="button" value="Add"/> <input type="button" value="Delete"/></p>	Name	Log Level	Output To	<input type="checkbox"/> LogFatal2Sys	Fatal	System Log	<input type="checkbox"/> LogAll2File	All Log Levels	File
Name	Log Level	Output To								
<input type="checkbox"/> LogFatal2Sys	Fatal	System Log								
<input type="checkbox"/> LogAll2File	All Log Levels	File								
Scope File Name*	<input type="text" value="/oblix/logs/scopefile.lst"/>									
SNMP State*	<input checked="" type="radio"/> Off <input type="radio"/> On									
SNMP Agent Registration Port*	<input type="text"/>									

Note: If you change the fields marked with an asterix(*), you must restart this COREid Server.

2. Set the file auditing and database auditing flags according to your preference, and change whichever audit file attributes you prefer. Click save to put your changes into effect.
3. Repeat this for all COREid Servers in your NetPoint system, then proceed to: “To modify audit output formatting for the COREid system” on page 440.

To modify audit output formatting for the COREid system

1. On the host for any COREid Server that will connect to the audit database, navigate to: COREid System Console > Common Configuration > Configure Master Audit Policy > Modify.

Configure Master Audit Policy.

Date Type	<input type="text" value="yyyy-mm-ddThh:mm:ssTZD"/>
Date Separator	<input type="text" value="-"/>
Message Format	<input type="text" value="\"/> %ob_datetime% - %ob_event% - %ob_operation% - %ob_serverid% - %ob_ip% - %ob_url% - %ob_target.uid% - %ob_app% - %ob_source.uid% - %ob_profileattrs% - %ob_auditapp%\"/> \""/>
Escape Character	<input type="text" value="\"/>
Record Separator	<input type="text" value="#"/>
Field Separator	<input type="text" value="~"/>

2. Click anywhere within the Message Format text box, press Control-A to select everything within the text box, even the contents that are obscured, then press Delete.
3. Into the empty text box, insert exactly what appears between the double quotes in the following string:

```
"%ob_datetime% - %ob_event% - %ob_operation% - %ob_serverid% - %ob_ip% - %ob_url% - %ob_target.uid% - %ob_app% - %ob_source.uid% - %ob_profileattrs% - %ob_auditapp%"
```

Do not include the double quotes in the text box, and do not add a semi-colon or line return to the end of the string.

4. If you prefer, modify the default values in the Date Type, Date Separator, Escape Character, Record Separator, and Field Separator fields. Remain aware that if you do change any of these values, you will need to reconfigure the Crystal report templates used to generate NetPoint Audit Reports.
5. Click Save. The new message format string and any other changes you made will display in the Configure Master Audit Policy page.
6. The new message format string applies across the COREid System, so you do not need to repeat the process for the other COREid Servers, but you do need to perform a similar procedure to set the format string for the Access system. See the procedure "To modify audit output formatting for the Access system" on page 448.
7. Proceed to: "To specify global COREid system events and profile attributes for audit" on page 441.

To specify global COREid system events and profile attributes for audit

1. On any COREid Server host that will connect to the audit database, navigate to: COREid System Console > Common Configuration > Configure Global Audit Policies > Modify.

Modify Application Auditing Policy

Profile Attributes

----	▼
----	▼
----	▼
----	▼
----	▼
----	▼

Event name	Application auditing enabled	Audit success	Audit failure
Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Logout	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Password Management	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
License	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

2. Select up to 5 profile attributes to audit.
3. Modify the default audit flag settings for whichever common User, Group, and Organization Manager application events you prefer.
4. Click Save to apply these settings to all the COREid Servers in your system.
5. Proceed to: “To specify User Manager events for audit” on page 442.

To specify User Manager events for audit

1. On any COREid Server host that will connect to the audit database, navigate to: COREid System Console > User Manager Configuration > Configure Auditing Rules > Modify.

Modify Application Auditing Policy

Profile Attributes

----- ▾
 ----- ▾
 ----- ▾
 ----- ▾
 ----- ▾
 ----- ▾

Event name	Application auditing enabled	Audit success	Audit failure
Search	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
View Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Modify Profile	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View Location	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Modify Location	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Substitute Right	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Workflow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Configuration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Deactivated User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Reactivated User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Created User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Deleted User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Workflow Duration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Save Cancel

2. Select up to 5 profile attributes to audit.
3. Modify the default audit flag settings for whichever common User Manager application events you prefer. COREid Server
4. Click Save to apply these settings to all the COREid Servers in your system.
5. Proceed to: “To specify Group Manager events for audit” on page 443.

To specify Group Manager events for audit

1. On any COREid Server host that will connect to the audit database, navigate to: COREid System Console > Group Manager Configuration > Configure Auditing Rules > Modify.

Profile Attributes

----- ▾
----- ▾
----- ▾
----- ▾
----- ▾
----- ▾

Event name	Application auditing enabled	Audit success	Audit failure
Search	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
View Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Modify Profile	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View My Group	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
View Group Member	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Expand Group	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Subscribe Group	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Workflow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Configuration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Workflow Duration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

2. Select up to 5 profile attributes to audit.
3. Modify the default audit flag settings for whichever common Group Manager application events you prefer.
4. Click Save to apply these settings to all the COREid Servers in your system.
5. Proceed to: “To specify Organization Manager events for audit” on page 444.

To specify Organization Manager events for audit

1. On any COREid Server host that will connect to the audit database, navigate to: COREid System Console > Organization Manager Configuration > Configure Auditing Rules > Modify.

Modify Application Auditing Policy

Profile Attributes

----	▼
----	▼
----	▼
----	▼
----	▼
----	▼

Event name	Application auditing enabled	Audit success	Audit failure
Search	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
View Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Modify Profile	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Containment Profile	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Container Limit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View Location	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Modify Location	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Workflow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Configuration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Workflow Duration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

2. Select up to 5 profile attributes to audit.
3. Modify the default audit flag settings for whichever common Organization Manager application events you prefer.
4. Click Save to apply these settings to all the COREid Servers in your system.
5. Proceed to: “To verify that all COREid Servers can record data to the audit database (Windows)” on page 444.

To verify that all COREid Servers can record data to the audit database (Windows)

1. From any page within the COREid Console of any COREid Server for which you have completed all the audit setup procedures up to this point, click Logout in the upper right corner of the application window.
2. Click OK when asked if you really want to log out.

- Open the SQL Server Query Analyzer window on the machine hosting your audit base. (You minimized this window when you completed the procedure “To verify the audit schema in (SQL Server/Windows)” on page 427.)

If, for any reason, the window is no longer open, re-launch it by navigating to: Start > Programs > Microsoft SQL Server > Query Analyzer > File > Open > *Login_Credentials* > OK > File > Open > *audit_sql_path* > OK

where *Login_Credentials* is the user name and password you specified when installing SQL Server and *audit_sql_path* is the path to the audit.sql file you copied to the audit database host and subsequently modified in the procedure “To verify the audit schema in (SQL Server/Windows)” on page 427.

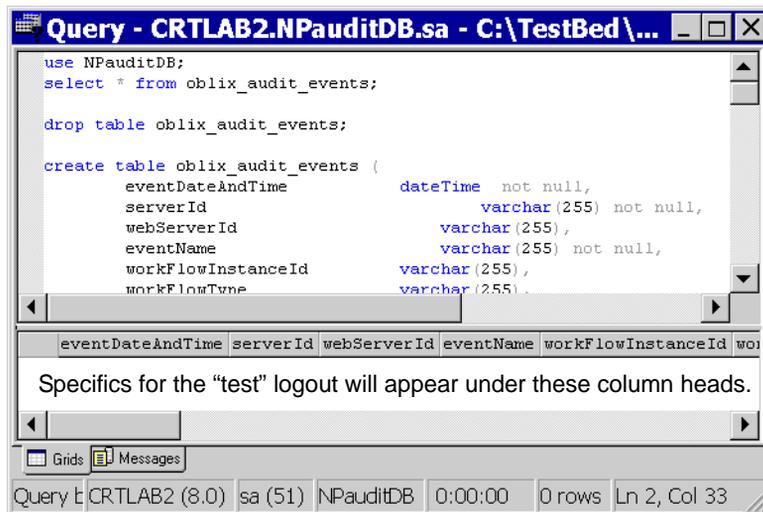
- Press F5 to execute audit.sql.

You previously saved audit.sql after adding the following lines:

```
use AuditDBName;
use select * from oblix_audit_events;
```

where *AuditDBName* specifies the NetPoint audit database you created in the procedure “To create the NetPoint audit database using SQL Server (Windows only)” on page 360.

The column headings for the NetPoint schema appear at the bottom of the Query window with particulars for the logout under the appropriate columns.



- Proceed to: “To enable and configure auditing for each Access Server” on page 447.

To verify that all COREid Servers can record data to the audit database (Unix)

1. From any page within the COREid Console of any COREid Server for which you have completed all the audit setup procedures up to this point, click Logout in the upper right corner of the application window.
2. Click OK when asked if you really want to log out.
3. From the mysql prompt, enter the following command:

```
mysql> Select * from oblix_audit_events order by  
eventDateAndTime;
```

The last row produced by this query should be the logout event you executed with an appropriate time stamp and other details.

4. Proceed to: “To enable and configure auditing for each Access Server” on page 447.

To enable and configure auditing for each Access Server

1. On any Access Server you plan to connect to the audit database, navigate to: Access System Console > Access System Configuration > Access Server Configuration > *ServerName* > Modify

where *ServerName* specifies the Access Server you want to modify.

Modify Access Server

Name	aaa-brass-7020
Hostname*	<input type="text" value="brass"/>
Port*	<input type="text" value="7020"/>
Debug*	<input checked="" type="radio"/> Off <input type="radio"/> On
Debug File Name*	<input type="text"/>
Transport Security*	<input type="radio"/> Open <input checked="" type="radio"/> Simple <input type="radio"/> Cert
Maximum Client Session Time (hours)*	<input type="text" value="24"/>
Number of Threads*	<input type="text" value="60"/>
Access Management Service*	<input type="radio"/> Off <input checked="" type="radio"/> On
Audit to Database (on/off)*	<input checked="" type="radio"/> Off <input type="radio"/> On
Audit to File (on/off)*	<input checked="" type="radio"/> Off <input type="radio"/> On
Audit File Name	<input type="text"/>
Audit File Size (bytes)	<input type="text" value="0"/>
Buffer Size (bytes)	<input type="text" value="512000"/>
File Rotation Interval (seconds)	<input type="text" value="0"/>
Engine Config Refresh Period (seconds)	<input type="text" value="14400"/>
URL Prefix Reload Period (seconds)	<input type="text" value="7200"/>
Password Policy Reload Period (seconds)	<input type="text" value="7200"/>
Maximum Elements in User Cache*	<input type="text" value="100000"/>
User Cache Timeout (seconds)*	<input type="text" value="1800"/>
Maximum Elements in Policy Cache *	<input type="text" value="10000"/>
Policy Cache Timeout (seconds)*	<input type="text" value="7200"/>
SNMP State*	<input checked="" type="radio"/> Off <input type="radio"/> On
SNMP Agent Registration Port*	<input type="text"/>

Please note that if you change the fields marked with an asterisk(*), you will have to restart this Access Server.

Update Cache

2. Set the file auditing and database auditing flags according to your preference.
3. Change whichever audit file attributes you prefer, then click Save to commit your changes.

If you change any of the attributes marked with asterisks, you must restart your Access Server to make the changes take effect.

4. Repeat this for all Access Servers in your NetPoint system, then proceed to: “To modify audit output formatting for the Access system” on page 448.

To modify audit output formatting for the Access system

1. On any Access Server you plan to connect to the audit database, navigate to: Access System Console > Access System Configuration > Common Information Configuration > Master Auditing Rule > Add (or Modify).

Add the Master Audit Rule

Profile Attributes

Audit Events
Authentication Success
Authentication Failure
Authorization Success
Authorization Failure

Audit Event Mapping
Authentication Success maps to AUTHN_SUCCESS
Authentication Failure maps to AUTHN_FAIL
Authorization Success maps to AUTHZ_SUCCESS
Authorization Failure maps to AUTHZ_FAIL

Audit Date Type mm/dd/yyyy Format

Audit Escape Character \

Audit Record Format
%ob_datetime% - %ob_event% - %ob_operation% - %ob_serverid% - %ob_ip% - %ob_url% - %ob_userid% - %ob_time_no_offset% - %ob_resrc_scheme% - %ob_wgid% - %ob_wgcontext% - %ob_reason%

Update Cache

Save Cancel

2. Click anywhere within the Audit Record Format text box, press Control-A to select everything within the text box, even the contents that are obscured, then press Delete.
3. Into the empty text box, insert exactly what appears between the double quotes in the following string:

```
“%ob_dateti me% - %ob_event% - %ob_operati on% - %ob_serveri d% - %ob_i p% - %ob_url% - %ob_useri d% - %ob_ti me_no_offset% - %ob_resrc_scheme% - %ob_wgi d% - %ob_wgcontext% - %ob_reason%”
```

Do not include the double quotes in the text box, and do not add a semi-colon or line return to the end of the string.

4. In the Profile Attributes box, type the name of a profile attribute you want to audit, then click the plus sign (+) to the right of the text box. Repeat this step to add other profile attributes.
5. Select the events you want to audit.
 - If you prefer, modify the default event mappings.
 - If you prefer, modify the default values in the Audit Date Type and Audit Escape Character fields. Remain aware that if you do change any of these

values, you need to reconfigure the Crystal report templates used to generate NetPoint Audit Reports.

6. Click Save. The new message format string and any other changes you made appear in the Master Audit Rule page.
7. The new message format string applies across the Access System, so you do not need to repeat the process for the other Access Servers, but you do need to perform a similar procedure to replace the format string for the COREid system. See the procedure “To modify audit output formatting for the COREid system” on page 440.
8. Proceed to the procedure “To create and manage User access privilege reports” on page 449.

To create and manage User access privilege reports

1. On any Access Server you plan to connect to the audit database, navigate to Access System Console > System Management > Manage Reports > Add.

Add a new Report

User Access Privileges Report

Report Name

Description

Access Server

Results Storage
 Store in Database
 Store in File
Name of File

List of Resources

URL	Resource Type	Resource Operation
<input type="button" value="Add"/>		

From this IP Address

Date/Time of access
 Any
 Specific date and time

Date

Time : :

Timezone

Check access for the following user(s)
 selected users
 all users

2. In the Report Name field, type a descriptive name such as “Midnight Access.”
3. In the Description field, type a longer explanation of the report content, such as “Who has night shift access to the loading dock shipping manifest URLs.”

4. Specify whether to send the information to the audit database or the audit file on the local host. If you specify the audit file, you must provide a file name.
5. In the “From this IP Address field,” type the IP of the host for a specific web browser whose access you want to test.
6. In the “Date/Time...” field, select the date, time, and time zone for which you wish to test access. This can be a point in the future, because the audit feature does not actually report the historical results of a actual access attempt; rather, it consults the policy and profile information stored on the NetPoint directory server to calculate whether the specified users currently have permission to access the specified resource at the specified time.
7. Click the Add button near the List of Resources label to add URLs to the list of resources to be tested.

Add Resource Rule

URL http(s)://

Resource Type

Resource Operation

<input type="checkbox"/> GET	<input type="checkbox"/> POST	<input type="checkbox"/> PUT	<input type="checkbox"/> HEAD
<input type="checkbox"/> DELETE	<input type="checkbox"/> TRACE	<input type="checkbox"/> OPTIONS	<input type="checkbox"/> CONNECT
<input type="checkbox"/> OTHER			

8. Type the URL to be tested.
9. Set the Resource type to http or ejb.
10. Check the action(s) you want tested.
11. Click Save to return to the “Add a new report page.”
12. Click Add again to add another resource to be tested, or click “all users” or “specify users” to proceed to the next step.

The screenshot shows the NetPoint Selector interface. At the top, there's a search bar with 'Full Name' selected, 'That Contains' as the operator, and 'valiv' as the search term. Below the search bar, there are 'Done' and 'Cancel' buttons. The main area is divided into two sections: 'Search Results' on the left and 'Selected' on the right. The 'Search Results' section shows a table with one entry: 'Rohit Valiveti'. The 'Selected' section shows a table with one entry: 'Rohit Valiveti'. There are 'Delete All' and 'Add All' buttons in the 'Selected' section. The interface also includes 'Table View' and 'Custom View' options, and 'Previous' and 'Next' navigation buttons.

13. If you clicked “specify users,” type into the appropriate field of the Selector, a unique string from the name of a user you want to test, then click Go. (See the preceding screen shot for an example.)

- a) Click the Add button next to the name you want to add to the list.
 - b) Repeat this step until you have added all the users you want to test, then click Done.
14. After the Add a new Report page reappears, click Save to commit your changes.

Setting up NetPoint Audit Reports

To make use of the preconfigured Crystal Reports templates supplied with NetPoint, you must install the bundled Crystal Reports application on a Windows machine within your NetPoint server domain. (Crystal Reports cannot be installed on Unix machines, but it can make use of information in a database generated by MySQL installed on a Unix machine.)

In addition to installing Crystal Reports 9, you must also install a patch.

The NetPoint server installation directories are installed with certain templates, sample reports, database schema, and database drivers which are used by the Crystal Reports application. These are distinct from the Crystal Reports software itself. You must copy them from a NetPoint server install directory to the machine hosting your Crystal Reports software.

Task overview: To set up NetPoint Audit Reports

1. Install Crystal Reports 9.22a on a Windows machine that can connect to the machine hosting SQL Server or MySQL.
2. Install the mandatory patch for Crystal Reports 9.
3. Copy the NetPoint audit report templates, the Oblix Crystal Repository, and associated resources to the machine hosting Crystal Reports.
4. Connect Crystal Reports to the NetPoint audit database by creating an ODBC data source definition and editing `orMap.ini`.
5. Connect Crystal Reports to the Oblix Crystal database by creating an ODBC data source definition and editing `orMap.ini`.

To install Crystal Reports

1. Obtain a copy of the Crystal Reports 9.22 installation package from the supplemental NetPoint 7 CD or by downloading it from the web site where you obtained your NetPoint 7 installation package.
2. Launch `setup.exe` and follow the prompts.
3. Specify whichever installation directory you prefer.

4. When prompted, enter the product key, which is provided with the purchase of the COREid Reporting package.
5. When prompted, specify “typical” for the installation type.
6. Proceed to “To install the patch for Crystal Reports” on page 452.

To install the patch for Crystal Reports

1. Download the Crystal Reports 9 patch from the following web site:

http://support.businessobjects.com/communityCS/FilesAndUpdates/cr90dbexwin_en.zip.asp

2. Unzip cr90dbexwinen.zip into a temporary folder on your hard disk, then launch CR90DBEXWIN_EN_200403.EXE.
3. Follow the prompts to complete the patch installation.
4. Proceed to the procedure “To copy the NetPoint-specific Crystal resources” on page 452.

To copy the NetPoint-specific Crystal resources

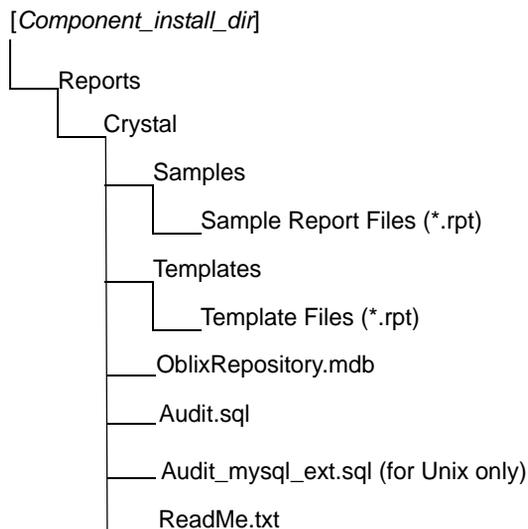
1. Using whatever methods you are comfortable with, copy the following resources from a NetPoint server installation to a directory of your choice on the machine hosting Crystal Reports.

Component_install_dir\oblix\reports

where *Component_install_dir* is the root installation directory for a NetPoint server connected to the audit database.

Make sure to copy everything in “.\reports” and its subdirectories. The following diagram shows the resources copied to the Crystal Reports machine.

Figure 16 Crystal Report resources installed for auditing to database



2. Proceed to the procedure “To connect Crystal Reports to the audit database” on page 453.

To connect Crystal Reports to the audit database

1. Follow the procedure described in “To create an ODBC data source definition (Windows)” on page 431 so that Crystal Reports can connect to the audit database.

Make sure that the DSN you specify and all associated details match exactly the values you specified for the RDBMS profile and the ODBC data source definitions you created for the NetPoint servers that connect to the audit database.

2. Proceed to the procedure “Task overview: To connect Crystal Reports to the Oblix Repository” on page 453.

Task overview: To connect Crystal Reports to the Oblix Repository

1. Create an ODBC data source definition to connect Crystal Reports to the Oblix/Crystal Repository (.mdb database).
2. Edit orMap.ini to equate the Oblix Repository with the Crystal Repository.

To create an ODBC data source definition to connect Crystal Reports to the Oblix/Crystal Repository

1. Follow the general procedure described in “To create an ODBC data source definition (Windows)” on page 431 so that Crystal Reports can connect to the audit database.

Except where noted in the steps that follow, use the values specified in the original procedure.

2. When prompted for a database driver, select “Microsoft Access driver (.mdb).”
3. For the Name parameter, choose some self-explanatory name such as OblixRepositorySysDSN.
4. Proceed to: “To edit orMap.ini” on page 454.

To edit orMap.ini

1. On the machine hosting Crystal Reports, navigate to:

```
C:\Program Files\Common Files\Crystal Decisions\2.5\bin
```

2. Open the file orMap.ini in any plain text editor.
3. Replace the line “Crystal Repository=Crystal Repository” with the following:

```
Crystal Repository = repository_DSN
```

where *repository_DSN* is the System DSN you created for the OblixRepository.mdb file. We have been using OblixRepositorySysDSN in our example.

12 SNMP Monitoring

This chapter focuses on NetPoint network monitoring through the Simple Network Management Protocol (SNMP) and includes the following topics:

- “Prerequisites” on page 456
- “About NetPoint SNMP and Agents” on page 456
- “About the NetPoint MIB and Objects” on page 457
- “Enabling and Disabling SNMP Monitoring” on page 471
- “Setting Up SNMP Agent and Trap Destinations” on page 472
- “Changing SNMP Configuration Settings” on page 474
- “Logging for SNMP” on page 476
- “NetPoint SNMP Messages” on page 476
- “Discrepancies Between Netstat and SNMP Values” on page 482

Note: For information about installing SNMP, refer to the *NetPoint 7.0 Installation Guide*.

Prerequisites

You need to have a network management station (NMS) installed, and you should be familiar with how to upload and display network statistics gathered from a Management Information Base (MIB). This chapter describes the NetPoint MIB objects and the Object Identifiers (OIDs) for these objects. However, this chapter does *not* provide information on how to use these OIDs in your NMS to collect statistics. For such information, refer to the documentation for your NMS.

Note: SNMP monitoring is just one of many NetPoint features that collect and present various types of information pertaining to your NetPoint system. NetPoint logging, auditing, and other reporting features, are described elsewhere in this guide.

About NetPoint SNMP and Agents

The NetPoint Simple Network Management Protocol (SNMP) enables you to monitor component activity on the network that hosts your NetPoint system by collecting and displaying NetPoint server-related SNMP data on a network management station (NMS). SNMP statistics commonly include data such as:

- The hosts, routers, and servers on your network
- The number of requests being processed on a particular device
- Whether or not a particular device is running
- Whether requests were processed successfully

SNMP data is displayed on a network management station (NMS). The NMS is a workstation running a network management application such as HP OpenView. You configure the NMS to display network statistics in a useful way, for instance, as a graph to show simple network statistics or to show whether the number of requests a device is processing falls within a set of defined limits.

You can capture SNMP statistics for the COREid Server and the Access Server running on any supported platform. NetPoint supports SNMP polling and trapping. Polling collects information such as:

- The version number of a component
- Configuration status
- Connection status
- Statistics on actions the component has processed

Event traps include information such as:

- Component failure
- Event failure
- Connection status
- Failure to complete actions

Note: NetPoint supports version 2 of the SNMP protocol.

The NetPoint SNMP Agent

The Simple Network Management Protocol (SNMP) is an application-layer protocol that enables network devices to exchange information. By using SNMP-transported data (such as successful operations and failure conditions), administrators can monitor network performance and solve problems. The NetPoint SNMP Agent enables you to implement SNMP-based data collection for the NetPoint COREid Server and Access Server. The NetPoint SNMP Agent enables collection of information such as the number of successful authentications performed by the Access Server and the number of requests processed by the COREid Server.

The NetPoint SNMP Agent is an optional installable component. The Agent collects information on the host where it is installed, so you must install an Agent on each host where you want to collect SNMP data. If installed, the Agent accesses information about the COREid or Access Server resident on the same server host on which the Agent was installed. The Agent is installed in *SNMP_install_dir*.

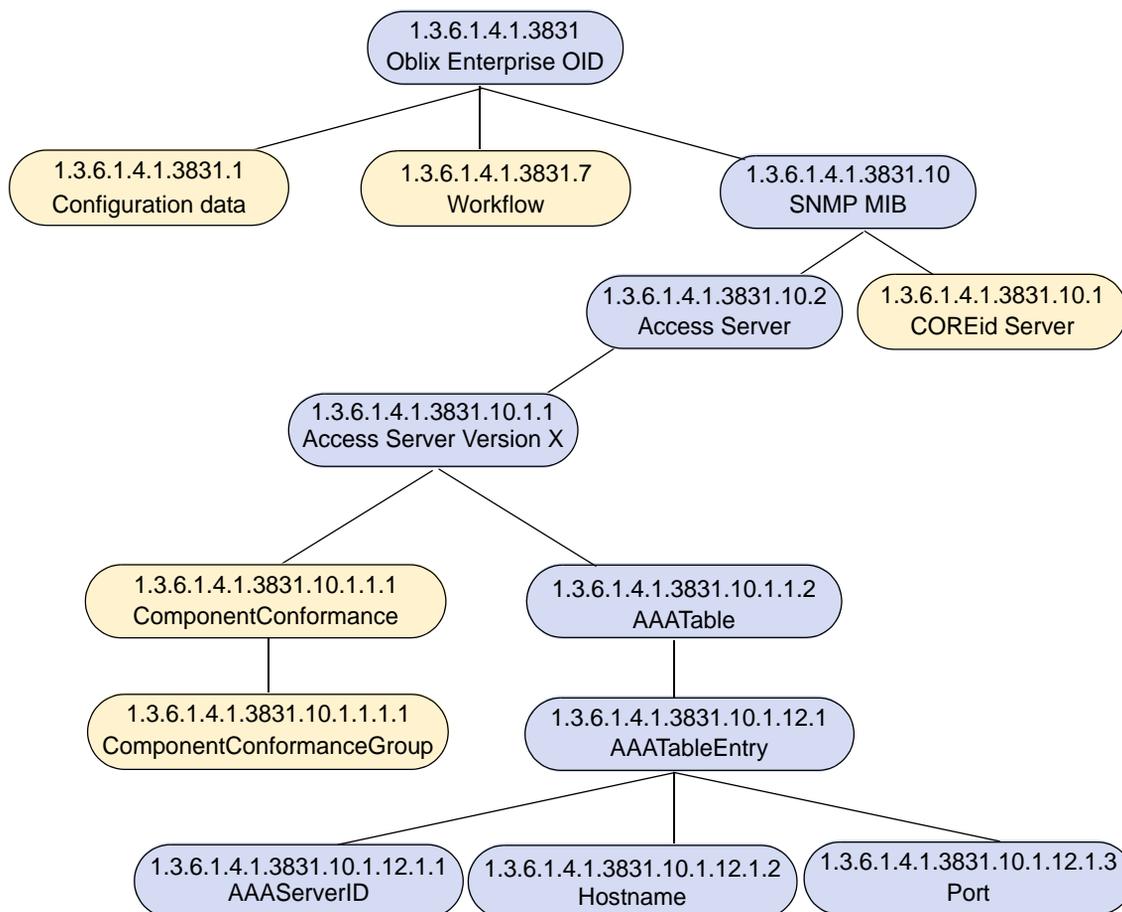
For information on installing the SNMP Agent, see the *NetPoint 7.0 Installation Guide*.

About the NetPoint MIB and Objects

The NetPoint Management Information Base (MIB) is a specification file that contains variables relevant to the status of different NetPoint components. The NetPoint SNMP Agent collects values for fields in the MIB.

Figure 17 illustrates the NetPoint MIB hierarchy.

Figure 17 The NetPoint MIB hierarchy



The NetPoint MIB can be expressed as a concatenation of branch and object identifiers (OIDs). The label from the MIB root to the top node of the NetPoint MIB is as follows:

i so . org . dod . i n t e r n e t . p r i v a t e . e n t e r p r i s e s . o b l i x . s n m p

MIB files are located in *SNMP_install_dir/oblix/mibs*. These files are conformant with SNMP Version 2.

The following discussions describe the MIB objects that are provided with the NetPoint SNMP component.

Note: Refer to your NMS documentation for information on uploading the MIB files to your NMS.

MIB Index Fields

Each MIB table contains one or more index fields. The index field values help you identify a unique row in the table.

For example, the index fields for `coreidInstanceTable` described in “COREid Server MIB Objects” on page 459 are `coreidHostname` and `coreidPort`. These entries are used as indexes because they uniquely identify a COREid installation. Suppose that you have two COREid Servers named `COREid1` and `COREid2`, each with a host name of `localhost` using ports 6023 and 6024, respectively. The indexes for these servers would be `localhost.6023` and `localhost.6024`.

To retrieve the first column value for `COREid1`, the object identifier you would request from the SNMP Agent would take the following logical form::

```
1.3.6.1.4.1.3831.10.1.1.2.1.1.localhost.6023
```

where `1.3.6.1.4.1.3831.10.1.1.2.1.1` signals that you want the first column of `coreidInstanceTable`, for the element with an index value of `localhost.6023`. The index is represented in numeric notation (similar to specifying an OID) which actually contains the length of the string followed by ascii codes for the characters in the string. As a result, this example:

```
1.3.6.1.4.1.3831.10.1.1.2.1.1.localhost.6023
```

is actually represented as follows:

```
1.3.6.1.4.1.3831.10.1.1.2.1.1.9.108.111.99.97.108.104.111.115.116.6023
```

Note: If you want the entire table to be returned in your SNMP requests, It is not necessary to know the values of the index fields.

COREid Server MIB Objects

Table 66 contains the COREid Server objects in the MIB. The path to this information is the following:

```
iso.org.dod.internet.private.enterprises.oblix.snmp.coreid.versionone
```

The name of this table is `coreidInstanceTable`. Its index fields are `coreidHostname` and `coreidPort`. It describes COREid Server instances.

Table 66 COREid Server MIB Objects

Managed Object	Syntax	Description
<code>coreidInstanceTable</code> OID: 1.3.6.1.4.1.3831.10.1.1.2	n.a.	Primary table name.

Table 66 COREid Server MIB Objects

Managed Object	Syntax	Description
coreidId OID: 1.3.6.1.4.1.3831.10.1.1.2.1.1	SnmpAdminString (size 0-255)	The identifier for the COREid Server instance.
coreidHostname OID: 1.3.6.1.4.1.3831.10.1.1.2.1.2	SnmpAdminString (size 0-255)	The hostname of the machine on which this COREid Server runs. The hostname is an index for this table.
coreidPort OID: 1.3.6.1.4.1.3831.10.1.1.2.1.3	Integer (0-65535)	The port on which the COREid Server listens. The port number is an index for this table.
coreidMode OID: 1.3.6.1.4.1.3831.10.1.1.2.1.4	Integer (0-5)	The transport security mode between the COREid Server and WebPass. 0—Open 1—Simple 2—Cert
coreidStartTime OID: 1.3.6.1.4.1.3831.10.1.1.2.1.5	DateAndTime	The time when the COREid Server was last started.
coreidServiceThreads OID: 1.3.6.1.4.1.3831.10.1.1.2.1.6	Integer (0-65535)	The number of service threads in the COREid Server instance. The number of threads is set in the administration console. The parameter NumberOfServiceThreads in scoreboard_params.lst controls how many slots are allocated (using one per service thread) to maintain SNMP information for each service thread.
coreidNumOfLanguagesConfigured OID: 1.3.6.1.4.1.3831.10.1.1.2.1.7	Integer (0-65535)	The number of languages installed for this COREid Server instance.
coreidNumOfLogins OID: 1.3.6.1.4.1.3831.10.1.1.2.1.8	counter64	The number of successful logins to the COREid Server instance.
coreidNumOfLoginsFailure OID: 1.3.6.1.4.1.3831.10.1.1.2.1.9	Counter64	The number of failed login attempts to the COREid Server instance.

Table 66 COREid Server MIB Objects

Managed Object	Syntax	Description
coreidRequestsProcessed OID: 1.3.6.1.4.1.3831.10.1.1.2.1.10	Counter64	The number of requests processed by the COREid Server instance.
coreidNumOfRequestsSuccess OID: 1.3.6.1.4.1.3831.10.1.1.2.1.11	Counter64	The number of requests successfully handled by this COREid Server instance.
coreidNumOfRequestsFail OID: 1.3.6.1.4.1.3831.10.1.1.2.1.12	Counter64	The number of requests for this COREid Server that produced an error.
coreidTotalServiceTime OID: 1.3.6.1.4.1.3831.10.1.1.2.1.13	Counter64	Total time, in nanoseconds, the COREid Server has taken to serve requests since the last restart?
coreidTotalNumOfCacheFlush RequestSuccess OID: 1.3.6.1.4.1.3831.10.1.1.2.1.14	Counter64	Total number of successful cache flush requests issued by the COREid Server.
coreidTotalNumOfCacheFlush RequestFail OID: 1.3.6.1.4.1.3831.10.1.1.2.1.15	Counter 64	Total number of unsuccessful cache flush requests issued by the COREid Server
coreidNumOfPluginsLoaded OID: 1.3.6.1.4.1.3831.10.1.1.2.1.16	Counter64	The number of plug-ins loaded by the COREid Server instance.
coreidNumOfEmailSentFail OID: 1.3.6.1.4.1.3831.10.1.1.2.1.17	Counter64	The number of failed attempts to send email from this COREid Server instance.

Table 66 COREid Server MIB Objects

Managed Object	Syntax	Description
coreidOverflowFlagDirectory ServerSlots OID: 1.3.6.1.4.1.3831.10.1.1.2.1.18	Integer (0-65535)	A flag indicating that the number of configured SNMP information slots for the directory server was insufficient. The variable NumberOfConfiguredDS in scoreboard_params.lst defines the number of slots, using one slot per directory server. If the value of NumberOfConfiguredDs is less than the actual number of directories that the COREid Server has contacted, the value for coreidOverflowFlagDirectory ServerSlots is set to 1. This flag only indicates an overflow condition. It does not convey how many slots are missing.
coreidOverflowForPPPActionsSlots OID: 1.3.6.1.4.1.3831.10.1.1.2.1.19	Integer (0-65535)	The number of "hooked up" Identity Event API plug-in actions for which a slot could not be allocated.

Table 67 contains the MIB objects for capturing information about the Identity Event API plug-in, which allows you to create external events for workflows. More information about this plug-in is provided in the *NetPoint 7.0 Developer Guide*. This table has three index fields: coreidHostname, coreidPort, and pppRowIndex. The path to this information is the following:

iso.org.dod.internet.private.enterprises.oblix.snmp.coreid.versionone.
pppActionsTable

Table 67 Identity Event API MIB Objects

Managed Object	Syntax	Description
pppActionsTable	n.a.	Primary table name.
pppRowIndex OID: 1.3.6.1.4.1.3831.10.1.1.3.1.1	Integer (0 - 65535)	This field is used only for indexing purposes. This value, along with its parent index values, forms a unique identifier for the row.
pppActionName OID: 1.3.6.1.4.1.3831.10.1.1.3.1.2	SnmpAdminString (size 0-255)	The name of the PPP action.

Table 67 Identity Event API MIB Objects

Managed Object	Syntax	Description
pppFunctionName OID: 1.3.6.1.4.1.3831.10.1.1.3.1.3	SnmpAdminString (size 0-255)	The name of the external function that is executed for the given hook.
pppPluginPath OID: 1.3.6.1.4.1.3831.10.1.1.3.1.4	SnmpAdminString (size 0-255)	The path for the PPP plug-in.
totalCount OID: 1.3.6.1.4.1.3831.10.1.1.3.1.5	Counter64	The total number of times the PPP action is executed.
pppOKCount OID: 1.3.6.1.4.1.3831.10.1.1.3.1.6	Counter64	The number of times that the return code STATUS_PPP_OK is received for this PPP action.
pppAbortCount OID: 1.3.6.1.4.1.3831.10.1.1.3.1.7	Counter64	The number of times that the return code STATUS_PPP_ABORT is received for this PPP action.
pppWorkflowRetryCount OID: 1.3.6.1.4.1.3831.10.1.1.3.1.8	Counter64	The number of times that the return code STATUS_PPP_WF_RETRY is received for this PPP action.
pppWorkflowAsyncCount OID: 1.3.6.1.4.1.3831.10.1.1.3.1.9	Counter64	The number of times the return code STATUS_PPP_WF_ASYNC is received for this PPP action.

The following table contains information about the directory server that communicates with the COREid Server. This table has three index fields: coreidHostname, coreidPort, and coreidDSRowIndex. The path to this information is the following:

iso.org.dod.internet.private.enterprises.oblix.snmp.coreid.versionone.coreidDirectoryServerTable

Table 68 COREid Directory Server MIB Objects

Managed Object	Syntax	Description
coreidDirectoryServerTable	n.a.	Primary table name.
coreidDSRowIndex OID: 1.3.6.1.4.1.3831.10.1.1.4.1.1	Integer (0-65535)	This field is used for indexing purposes only. This value, along with its parent index values, forms a unique identifier for the row.
coreidDirectoryServerHost name OID: 1.3.6.1.4.1.3831.10.1.1.4.1.2	SnmpAdminString (size 0 - 255)	The hostname of the directory server.

Table 68 COREid Directory Server MIB Objects

Managed Object	Syntax	Description
coreidDirectoryServerPort OID: 1.3.6.1.4.1.3831.10.1.1.4.1.3	Integer (0-65535)	The directory server port.
coreidDirectoryServerMode OID: 1.3.6.1.4.1.3831.10.1.1.4.1.4	Integer (0-65535)	The directory server communication mode: 0—Open 1—SSL
coreidDirectoryServerNoOfLiveConnections OID: 1.3.6.1.4.1.3831.10.1.1.4.1.5	Integer (0-65535)	The number of connections against the directory.

Table 69 contains the COREid objects in the MIB for system events that can be mapped to SNMP traps.

The SNMP Agent supports sending trap messages to multiple NMS systems. The path to this information is the following:

iso.org.dod.internet.private.enterprises.oblix.snmp.coreid.versionone

For example, the full path to the oblixCoreidServerDown trap is the following:

iso.org.dod.internet.private.enterprises.oblix.snmp.coreid.versionone.oblixCoreidServerDown

Table 69 COREid Server Traps

Managed Object	Fields sent with the trap	Description
oblixCoreidServerDown OID: 1.3.6.1.4.1.3831.10.1.1.0.7001	coreidId coreidHostname coreidPort	A trap generated when the SNMP Agent detects that the COREid Server has done a shutdown with errors. This trap contains the server ID, host name, and port.
oblixCoreidServerStart OID: 1.3.6.1.4.1.3831.10.1.1.0.7002	coreidId coreidHostname coreidPort	This trap is generated when the SNMP Agent detects that the COREid Server has been started or restarted. This trap contains the server ID, host name, and port.
oblixCoreidServerFailure OID: 1.3.6.1.4.1.3831.10.1.1.0.7003	coreidId coreidHostname coreidPort	This trap is generated when the SNMP Agent detects that the COREid Server has not done a clean shutdown or has crashed. This trap contains the server ID, host name, and port.
oblixCOREidDSFailure OID: 1.3.6.1.4.1.3831.10.1.1.0.7004	coreidId coreidHostname coreidPort coreidDirectoryServer Hostname coreidDirectoryServer Port	This trap is generated when the COREid Server detects that the directory server that it is connected to is down.

Access Server MIB Objects

Table 70 describes the Access Server SNMP objects that are available through the MIB. The path to this information is the following:

i so. org. dod. i nternet. pri vate. enterpri ses. obl i x. snmp. aaa.
versi one

Table 70 Access Server MIB Objects

Managed Object	Syntax	Description
aaalInstanceTable OID: 1.3.6.1.4.1.3831.10.2.1.2	n.a.	Primary table name.

Table 70 Access Server MIB Objects

Managed Object	Syntax	Description
aaald OID: 1.3.6.1.4.1.3831.10.2.1.2.1.1	SnmpAdminString (size 0-255)	The identifier for this Access Server instance, as specified in the Access System Console.
aaaHostname OID: 1.3.6.1.4.1.3831.10.2.1.2.1.2	SnmpAdminString (size 0-255)	The name of the machine where the Access Server was installed, as specified in the Access System Console. The host name is an index for this table.
aaaPort OID: 1.3.6.1.4.1.3831.10.2.1.2.1.3	Integer (0-65535)	The port on which the Access Server listens. The port number is an index for this table.
aaaMode OID: 1.3.6.1.4.1.3831.10.2.1.2.1.4	Integer (0-65535)	The transport security mode between the Access Server and other NetPoint components. 0—Open 1—Simple 2—Cert
aaaNoOfQueues OID: 1.3.6.1.4.1.3831.10.2.1.2.1.5	Integer (0-65535)	The number of service queues for this Access Server instance.
aaaThreadsPerQueue OID: 1.3.6.1.4.1.3831.10.2.1.2.1.6	Integer (0-65535)	The number of threads per service queue for this Access Server instance.
aaaNoOfListenerThreads OID: 1.3.6.1.4.1.3831.10.2.1.2.1.7	Integer (0-65535)	The number of listener threads spawned. There will be one thread per WebGate-Access Server connection.
aaaNoofConnectionWatcherThreads OID: 1.3.6.1.4.1.3831.10.2.1.2.1.8	Integer (0-65535)	The number of LDAP connection watcher threads.
aaaOverflowFlagDirectoryServerSlots OID: 1.3.6.1.4.1.3831.10.2.1.2.1.9	Integer (0-65535)	A flag indicating whether there are insufficient slots for the number of directories configured for the Access Server. This means that the administrator needs to update the file <i>install_dir/access/oblix/config/obscoreboardparams.lst</i> . 0 - No overflow 1 - Overflow occurred

Table 70 Access Server MIB Objects

Managed Object	Syntax	Description
aaaOverflowForAuthentication PluginSlots OID: 1.3.6.1.4.1.3831.10.2.1.2.1.10	Integer (0-65535)	The number of authentication plug-ins whose information could not be displayed. The administrator needs to update the <i>install_dir/access/oblix/config/obscoreboardparams.lst</i> file.
aaaOverflowForAuthorization PluginSlots OID: 1.3.6.1.4.1.3831.10.2.1.2.1.11	Integer (0-65535)	The number of authorization plug-ins whose information could not be displayed. The administrator needs to update the <i>install_dir/access/oblix/config/obscoreboardparams.lst</i> file.
aaaTimeAuditLogWasRotated OID: 1.3.6.1.4.1.3831.10.2.1.2.1.12	DateAndTime	Time when the audit log file was rotated. This setting is determined in the configuration for this Access Server specified in the Access System Console.
aaaStartTime OID: 1.3.6.1.4.1.3831.10.2.1.2.1.13	DateAndTime	The date and time when this Access Server instance was last started.
aaaAuthenticationsSuccess OID: 1.3.6.1.4.1.3831.10.2.1.2.1.14	Counter64	The number of successful authentications by the Access Server instance.
aaaAuthenticationsSuccess OID: 1.3.6.1.4.1.3831.10.2.1.2.1.15	Counter64	The number of successful authentications by this Access Server instance.
aaaAuthenticationsDenied OID: 1.3.6.1.4.1.3831.10.2.1.2.1.16	Counter64	The number of unsuccessful authentications by this Access Server instance.
aaaAuthorizationsSuccess OID: 1.3.6.1.4.1.3831.10.2.1.2.1.17	Counter64	The number of successful authorizations by this Access Server instance.
aaaAuthorizationsDenied OID: 1.3.6.1.4.1.3831.10.2.1.2.1.18	Counter64	The number of unsuccessful authorizations by this Access Server instance.
aaaAuditRequests OID: 1.3.6.1.4.1.3831.10.2.1.2.1.19	Counter64	The number of audit requests made by this Access Server instance.

Table 71 is a sub-table of MIB objects that describe the directory server that communicates with the Access Server. This sub-table has index fields of aaaHostname, aaaPort, and aaaRowIndex. The path to this information is the following:

iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.versionone.aaaDirectoryServerTable

Table 71 Access System Directory Server MIB Objects

Managed Object	Syntax	Description
aaaDirectoryServerTable OID: 1.3.6.1.4.1.3831.10.2.1.3	n.a.	Primary table name.
aaaDSRowIndex OID: 1.3.6.1.4.1.3831.10.2.1.3.1.1	Integer (0-65535)	An index field. It does not contain any information.
aaaDirectoryServerHostname OID: 1.3.6.1.4.1.3831.10.2.1.3.1.2	SnmpAdminString (size 0-255)	The directory host name.
aaaDirectoryServerPort OID: 1.3.6.1.4.1.3831.10.2.1.3.1.3	Integer (0-65535)	The directory server port.
aaaDirectoryServerMode OID: 1.3.6.1.4.1.3831.10.2.1.3.1.4	Integer (0-65535)	The directory server communication mode with the Access Server: 0—Open 1—SSL
aaaDirectoryServerNoOfLiveConnections OID: 1.3.6.1.4.1.3831.10.2.1.3.1.5	Integer (0-65535)	The number of connections between the Access Server and the directory server.

Table 72 is a sub-table of MIB objects for capturing information on authentication plug-ins. This sub-table has index fields of aaaHostname, aaaPort, and authenticationPluginName. The path to this information is the following:

iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.
versionone.aaaauthenticationPluginsTable

Table 72 Authentication Plug-Ins MIB Objects

Managed Object	Syntax	Description
authenticationPluginsTable OID: 1.3.6.1.4.1.3831.10.2.1.4	n.a.	Primary table name.
authenticationPluginName OID: 1.3.6.1.4.1.3831.10.2.1.4.1.1	SnmpAdminString (size 0-255)	The name of the plug-in. The authentication plug-in name is an index for this table.
AuthenticationPluginPath OID: 1.3.6.1.4.1.3831.10.2.1.4.1.2	SnmpAdminString (size 0-255)	The path of the authentication plug-in.
AuthenticationPluginStatus OID: 1.3.6.1.4.1.3831.10.2.1.4.1.3	Integer (0-65535)	The status of the plug-in: 0—Not loaded 1—Loaded

The authorizationPluginsTable has index fields of aaaHostname, aaaPort, and authorizationPluginName. The path to this information is the following:

iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.
versionone.authorizationsPluginsTable

Table 73 Authorization Plug-Ins MIB Objects

Managed Object	Syntax	Description
authorizationPluginsTable OID: 1.3.1.4.1.3831.10.2.1.5	n.a.	Primary table name.
authorizationPluginName OID: 1.3.6.1.4.1.3831.10.2.1.5.1.1	SnmpAdminString (size 0-255)	The name of this plug-in.
AuthorizationPluginPath OID: 1.3.6.1.4.1.3831.10.2.1.5.1.2	SnmpAdminString (size 0-255)	The path of the authorization plug-in.
AuthorizationPluginStatus OID: 1.3.6.1.4.1.3831.10.2.1.5.1.3	Integer (0-65535)	The status of the plug-in: 0—Not loaded 1—Loaded

Table 74 is a sub-table that describes the number of requests in the queue for the Access Server. This table has indexes of aaaHostname, aaaPort, and aaaRequestQueueNumber. The path to this information is the following:

iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.
versionone.requestQueueInfoTable

Table 74 Request Queue MIB Objects

Managed Object	Syntax	Description
requestQueueInfoTable OID: 1.3.6.1.4.1.3831.10.2.1.5	n.a.	Primary table name.
aaaRequestQueueNumber OID: 1.3.6.1.4.1.3831.10.2.1.6.1.1	Integer (0-65535)	Index for the request queue.
aaaRequestQueueSize OID: 1.3.6.1.4.1.3831.10.2.1.6.1.2	Integer (0-65535)	The number of requests in the queue.

Table 75 contains objects in the MIB for system events that can be mapped to SNMP traps. The SNMP Agent supports sending trap messages to multiple NMS systems. The path to this information is the following:

iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.
versionone

For example, to add the full path to the oblixAAAServerDown trap, you would specify:

iso.org.dod.internet.private.enterprises.oblix.snmp.aaa.versionone.
oblixAAAServerDown

Table 75 Access Server Traps

Managed Object	Fields Sent with the Trap	Description
oblixAAAServerDown OID: 1.3.6.1.4.1.3831.10.2.1.0.7001	aaald aaaHostname aaaPort	A trap generated when the SNMP Agent detects that the Access Server has done a clean shutdown. This trap captures the Access Server ID, host name, and port.
oblixAAAServerStart OID: 1.3.6.1.4.1.3831.10.2.1.0.7002	aaald aaaHostname aaaPort	A trap generated whenever the Access Server is restarted. This trap captures the Access Server ID, host name, and port. The trap is generated immediately, so the time of the restart is the time of the trap generation.

Table 75 Access Server Traps

Managed Object	Fields Sent with the Trap	Description
oblixAAAServerFailure OID: 1.3.6.1.4.1.3831.10.2.1.0.7003	aaald aaaHostname aaaPort	A trap generated when the SNMP Agent detects that the Access Server has not done a shutdown with errors or has crashed. This trap captures the Access Server ID, host name, and port.
oblixAAADSFailure OID: 1.3.6.1.4.1.3831.10.2.1.0.7004	aaald aaaHostname aaaPort aaaDirectoryServer Hostname aaaDirectoryServerPort	A trap generated when the Access Server detects that the directory server it is connected to is down.

Enabling and Disabling SNMP Monitoring

You must explicitly configure NetPoint to enable collection of SNMP statistics. NetPoint provides options in the COREid and Access Servers to indicate that you want to enable SNMP and to indicate the TCP/IP port where contact will be established with the SNMP Agent.

Note: NetPoint does *not* provide a configuration setting for a polling interval to retrieve SNMP statistics. However, most NMS systems provide a polling configuration parameter. This parameter is used by the NMS to periodically poll the Agent to retrieve MIB values.

The following procedure describes how to start and stop the NetPoint SNMP Agent, and how to start the Agent on another port.

To configure NetPoint to collect SNMP statistics

1. From the COREid (or Access System Console), select System Configuration > Configure COREid Server (or Configure Access Server.)

For example:

COREid System Console > System Configuration > Configure COREid Server

2. Click a link for a particular server.
3. Select the Modify button to display the page where you can turn SNMP monitoring on or off, as follows:
 - **Turn On**—Select the SNMP State On button at the bottom of the page.

- **Turn Off**—Select the SNMP State Off button at the bottom of the page.
4. In the SNMP Agent Registration Port field, enter the port number to define or change the port on which the SNMP Agent listens.
 5. Restart the COREid Server (or Access Server).

Setting Up SNMP Agent and Trap Destinations

You use the following command to setup an SNMP Agent against an SNMP Manager:

```
setup_agent -i
```

The `-i` option is required.

Following procedures describe and illustrate how to configure the NetPoint SNMP Agent and trap destinations.

To configure the NetPoint SNMP Agent and trap destinations

1. Change to the directory containing the SNMP `setup_agent` command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

where `SNMPDIR` is the directory where you have installed the SNMP Agent.

2. Use the `setup_agent` command with the following options:

```
-i <install_dir>  
-g Configure General Parameters  
-u <Agent SNMP UDP Port>  
-c <Agent Community String>  
-p <Agent TCP Port>  
-S <Run in silent mode>  
--help Prints help message
```

To add a trap destination in silent mode

1. Change to the directory containing the SNMP `setup_agent` command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

2. Use the `setup_agent` command with the following options:

```
-a  
-m <Manager Station>  
-t <Trap port>
```

To delete an existing trap destination in silent mode

1. Change to the directory containing the SNMP `setup_agent` command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

2. Use the `setup_agent` command with the following options:

```
-d  
-m <Manager Station>  
-t <Trap port>
```

To configure general parameters first

1. Change to the directory containing the SNMP `setup_agent` command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

2. Use the `setup_agent` command as shown below.

For example:

```
> ./setup_agent -i $SNMPDIR -g -u <UDP Port> -c public -p <TCP Port>
```

This goes to the Manager Station Trap Configuration menu.

To add an SNMP Manager directly after general parameters

1. Change to the directory containing the SNMP `setup_agent` command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

2. Use the `setup_agent` command as shown below.

For example:

```
> ./setup_agent -i $SNMPDIR -a -m <Mgr M/c> -t <Mgr Port>
```

To delete an SNMP Manager directly after adding one

1. Change to the directory containing the SNMP `setup_agent` command.

For example:

```
> cd $SNMPDIR/oblix/tools/setup
```

2. Use the `setup_agent` command as shown below.

For example:

```
> ./setup_agent -i $SNMPDIR -d -m <Mgr M/c> -t <Mgr Port>
```

You can add any number of Manager Stations. The Agent then sends all the traps to the configured SNMP Managers.

Changing SNMP Configuration Settings

A configuration file named `obscoreboard_params.xml` or `.lst` contains information that defines the collection of SNMP statistics. This file is located in:

```
Component_install_dir/identity|access/oblix/config
```

where *Component_install_dir* is the directory where the component is installed and *identity|access* represents either the COREid System or Access System, respectively.

COREid System File—`obscoreboardparams.xml`

Access System File—`obscoreboardparams.lst`

In this file, you can configure threshold levels to determine when various MIB counters are activated.

The following parameters are specified only in the Access System file `obscoreboard_params.lst`:

- **NumberOfAuthenticationPlugins**—The maximum number of authentication plug-ins that may be loaded in the Access System. The Access Server maintains information on the number of plug-ins that are loaded. If the actual number of plug-ins loaded by the Access Server exceeds the value specified for `NumberOfAuthenticationPlugins`, the difference is displayed as the counter `aaaOverflowforAuthenticationPluginSlots`.
- **NumberOfAuthorizationPlugins**—The maximum number of authorization plug-ins that may be loaded in the Access System. The Access Server maintains information on the number of plug-ins that are loaded. If the actual number of plug-ins loaded by the Access Server exceeds the value specified for `NumberOfAuthorizationPlugins`, the difference is displayed as the counter `aaaOverflowforAuthorizationPluginSlots`.

The following parameter is specified only in the COREid file `obscoreboard_params.xml`:

- **NumberOfPPPPluginActions**—The number of Identity Event API plug-in actions that may be connected with this COREid Server. When the COREid Server starts, it reads this value and monitors the actual number of Identity Event API plug-ins. If the number of active plug-ins exceeds the value for

NumberOfPPPPluginActions, the difference is indicated by the counter coreidOverflowForPPPACTIONSLOTS.

The following parameters are provided in both scoreboard files:

- **NumberOfServiceThreads**—The value for this parameter is read by the COREid or Access Server at startup. This parameter controls how many slots to allocate (one per service thread) to maintain SNMP information for each service thread. The server monitors the number of service threads being used. The actual number of service threads is configured through the administration console, from the command line, or as part of a configuration file. This parameter does not control the number of threads to be started by the COREid Server. If the actual number used exceeds this value, there is no SNMP data generated regarding the extra threads.
- **NumberOfConfiguredDS**—The number of directory servers configured for this COREid or Access Server.
- **DsFailureTrapTimeSpan**—The amount of time to wait before sending the next failure trap to the same directory server.
- **NumOfSlotsInEventQueue**—The number of slots to be used in the event queue. This parameter must be updated if traps are not detected. However, the default value of five should be adequate for most installations.
- **SleepTimeInMilliSec**—The interval in milliseconds that the COREid or Access Server uses to check whether the NetPoint SNMP Agent is up and running.
- **semaphore_filepath**—Information about the semaphore created by the Access Server. Semaphores are used for synchronization between the NetPoint component (the COREid or Access Server) and the SNMP Agent. This information is used to automatically clean up the semaphore if the component crashes.
- **semaphore_id**—The Agent semaphore identifier.

Changing these settings affect the memory map file used for SNMP data collection. On Unix, the memory map file is located in `/tmp/netpoint/scoreboard/component/process-id.osb`. On Windows, this file is located in `Component_install_dir/oblix/scoreboard/process-id.osb`.

Logging for SNMP

The NetPoint SNMP Agent supports logging. Once the SNMP Agent is enabled, it is always set to a certain log level. The SNMP logs can assist with troubleshooting. You can configure what is logged and the type of logs to generate in the Agent configuration file. This file resides in

`SNMP_install_dir/oblix/config/snmp_agent_config_info.lst`

where `SNMP_install_dir` is the directory where the SNMP Agent was installed.

The `log_level` parameter in the Agent configuration file may have one of the following values:

- **0**—Debug
- **1**—Information
- **2**—Warning
- **3**—Error
- **4**—No logging (turns logging off)

NetPoint SNMP Messages

The following are SNMP-related messages.

Message—

`MErrNoConfigFile {Could not find agent configuration file at location (full path to the agent configuration file)}`

Description—The installation directory is not correct, or the configuration file is not present. Uninstall and reinstall the SNMP Agent.

Message—

`MLogAgentStarted {Agent successfully started on port SNMP port number}`

Description—Status message.

Message—

`MErrAddressInUse {Agent was not able to bind to port port number, address already in use}`

Description—The SNMP Agent is unable to bind to its configured TCP registration port. Reconfigure the Agent to use another TCP port, or make the port available by stopping the application using the port.

Note: If you change the Agent TCP registration port, you must also specify the new port when enabling SNMP for the COREid or Access Server using the appropriate System Console.

Message—

Agent was not able to bind to specified port, system lacked sufficient buffer space or queue was full.

Description—The NetPoint SNMP Agent port is unavailable.

Message—

MErrTLUnsupported {Agent was not able to bind to specified port, address family not supported by protocol family}

Description—The specified port does not support SNMP. Configure a different port.

Message—

MErrRetrieveIDs {Error: Unable to determine the uid/gid for which this snmp agent is installed.}

Description—The user who tried to start the SNMP Agent does not have the appropriate permissions. The user should start the SNMP Agent as root or as the user who installed the Agent.

Message—

MErrCouldNotSetIDs {Error: You don't have sufficient access rights to run this snmp agent.}

Description—You need to log in with administrative rights to be able to install the SNMP Agent. If you did not do this, the Agent is unable to run.

Message—

MLogAlreadyRunning {Agent is already running with process id (Process identifier of the agent).}

Description—The user is trying to start the Agent when it is already running.

Message—

MErrRegBindFailed {Error: Unable to bind to configured registration port (configured registration port number).}

Description—The SNMP Agent is unable to bind to the port configured on the NetPoint server configuration page. Specify a different port, as described in “Enabling and Disabling SNMP Monitoring” on page 471.

Message—

MErrRegListenFailed {Error: Unable to start listening on configured registration port (configured registration port number). }

Description—This message is displayed on Windows if the port is already in use by another application.

Message—

MErrReadingMsg {Error reading message sent by component. }

Description—The SNMP Agent and the NetPoint server talk over a TCP connection. If the Agent encounters a malformed message, it logs an error.

Message—

MErrNotRegMsg {Error: Agent expects only registration messages on the registration socket. }

Description—The Agent only expects registration messages on the TCP connection from a server that connects to it. If it finds that the message is not a registration message, it logs an error.

Message—

MErrMissingMmapFilename {Error: Registration message was missing the component scoreboard file name. }

Description—The scoreboard file is where the NetPoint server stores the statistics that are read by the Agent. This name is communicated by the server to the Agent at registration time. If the registration request is missing the file information, this message is logged.

Message—

MErrMappingScoreboard {Error: Unable to memory map the scoreboard file (full path to the scoreboard file) registered by component. }

Description—This error can occur due to file permission issues, that is, the Agent cannot read or open the scoreboard file.

Message—

MErrUnknownComponent {Error: Unknown component type specified in scoreboard file. }

Description—The component type is specified in the registration request. The Agent processes information for the COREid Server and Access Server. If the component type is not either of these, this message is logged.

Message—

MErrIndexExists {Error: A component has already registered in table (OID for the table for that component) with index (index that is already in use by some other component). }

Description—The same instance of a component tried to register again. Each instance of a component is uniquely identified by a key or index by the same SNMP Agent. If another component instance tries to register using the same key or index, this message is logged.

Message—

```
MErrCreatingAgentSemaphore {Error: Unable to create named semaphore (full path to the agent semaphore file) for agent-component event dispatching. }
```

Description—The Agent and the component create one semaphore that is cleaned up at shutdown. In case of unclean shutdown, the semaphores are deleted on the next server/Agent startup. Probable causes are that the system has run out of semaphores or there are permission issues while creating the semaphore.

Message—

```
MErrOnSelect {Error: Select() call returned error code (error code returned for the select() call). }
```

Description—This is an error code returned directly from the function. This message is used for troubleshooting purposes.

Message—

```
MErrOnPoll {Error: Poll() call returned error code (error code returned for the poll() call). }
```

Description—This is an error code returned directly from the function. This message is used for troubleshooting purposes.

Message—

```
MErrNotDeregMsg {Error: Agent expected a de-registration message on the socket, instead got a message with code (message code for the message received). }
```

Description—The Agent only expects a de-registration message from a component once the component has registered.

Message—

```
MErrRemovingComponent {Error: Component with table oid (OID for the table for that component) and index (index which identifies the component in that table) could not be removed. }
```

Description—The component has already de-registered, and there has been another request to remove it.

Message—

```
MErrMissingEvent {Error: Unable to retrieve event from component with table oid (OID for the table for that component) and index (index which identifies the component in that table). }
```

Description—The component sends an event to the Agent, and the Agent converts this to an appropriate trap. The component also signals the Agent that it has dispatched an event. If the Agent is signaled but it does not find an event, this message is logged.

Message—

```
MErrMissingTrapData {Error: Missing trap meta-data for component from table oid (OID for the table) and index (index that identifies the component in that table) with event (event identifier supplied by the component). }
```

Description—The component did not deliver the complete data for an event.

Message—

```
MLogMappedScoreboard {Mapped scoreboard file (full path to the scoreboard file) for a component. }
```

Description—This is a status message.

Message—

```
MLogComponentRegistered {Component registered with table oid (OID for the table) and index (index that identifies the component). }
```

Description—This is a status message.

Message—

```
MLogComponentDeregistered {Component with table oid (OID for the table) and index (index that identifies the component) de-registered. }
```

Description—This is a status message.

Message—

```
MLogComponentFailed {Component with table oid (OID for the table) and index (index that identifies the component) failed. }
```

Description—This is a status message indicating that the NetPoint component did not deregister properly. This action is treated as a component failure by the SNMP Agent.

Message—

MLogSentTrap {Sent trap with trap oid (OID for the trap sent) for component with table oid (OID for the component table) and index (index that identifies the component in the table).}

Description—This is a status message.

Message—

MLogSemCleanup {Found left-over semaphore from previous run with key (key for the stale left-over semaphore) and file path (file path for the stale left-over semaphore), successfully cleaned up the semaphore.}

Description—Status message. The Agent and the component create one semaphore that is cleaned up at shutdown. In case of unclean shutdown, the semaphores are deleted on the next server/Agent startup.

Message—

MErrSemCleanup {Found left-over semaphore with key (key for the stale left-over semaphore) and file path (file path for the stale left-over semaphore). Encountered errors while removing it.}

Description—The Agent and the component create one semaphore that is cleaned up at shutdown. In case of unclean shutdown, the semaphores are deleted on the next server/Agent startup. This message would be logged if the Agent encountered errors while cleaning up the semaphores from a previous run. There may be permission issues.

Message—

MSBCreateFailed {Access Server: Could not create scoreboard file (full path for the file) with size file size.}

Description—The probable cause for this message is the system could not create the file due to insufficient space.

Message—

MCreateSemFailed {Access Server: Could not create event queue semaphore with path full path.}

Description—The Agent and the component create one semaphore that is cleaned up at shutdown. In case of unclean shutdown, the semaphores are deleted on the next server/Agent startup. This message is generated when the system has run out of semaphores or there are permission issues when creating the semaphore. Try increasing the semaphore limit on the machine.

Message—

```
MSBDirCreateFailed {Access Server: Could not create scoreboard  
file file name. }
```

Description—The system could not create the required directory for the scoreboard file, probably due to insufficient permissions.

Discrepancies Between Netstat and SNMP Values

When using the netstat command, the value returned for this command may not always match the information collected for the MIB variables:

```
aaaDirectoryServerNoOfLiveConnections  
coreidDirectoryServerNoOfLiveConnections
```

Table 76 explains the reason for this discrepancy and the chain of events that takes place.

Table 76 Netstat Values and Number of Live Connections Displayed

Event	Number Of Live Connections	Netstat Value	Comments
Server startup followed by directory server access.	5	5	
The directory server goes down.	5	0	NetPoint does not update the counter unless it receives a request.
NetPoint tries to use a connection for accessing the directory server for servicing a request.	4	0	The directory server access returns an error because the directory server is down. The connection is marked as down and NetPoint decreases the NumberOfLiveConnections by one.
Directory server is restarted and NetPoint tries to reestablish the broken connection.	5	1	When a new connection is formed, NetPoint increments the NumberOfLiveConnections by one. The mismatch between NumberOfLiveConnections and the Netstat value will be seen until all of the remaining four connections are marked as down and new connections are formed. The status for the remaining four connections will not be visible unless they are used.
NetPoint re-establishes all of the broken connections.	5	5	The netstat value matches NumberOfLiveConnections only after all connections are formed.

Configuring the Shutdown Interval

To ensure that a NetPoint component can perform a clean shutdown, enough time must be allocated to ensure that all cleanup activities can be completed. For the COREid Server, the Access Server, and the SNMP Agent, the `shutdown_time` parameter specifies the time allocated for the server to attempt a clean shutdown. This parameter is located in `globalparams.lst` and `globalparams.xml`. The default shutdown time is five seconds.

The `globalparams` file location is as follows:

For Access Server:

```
AccessServer_install_dir/access/oblix/apps/common/bin/globalparams.lst
```

For COREid Server:

```
COREidr_install_dir/identity/oblix/apps/common/bin/globalparams.xml
```

The default shutdown time appears as follows in these files:

```
shutdown_time:5
```

You can change the value to any time, specified in seconds.

SECTION IV: APPENDICES AND INDEX

A Deploying NetPoint with Active Directory

After you complete the activities in the *NetPoint 7.0 Installation Guide* to install and set up NetPoint with Active Directory, you can complete activities here to configure these components for daily use and maintenance. For details, see:

- “Setting Up Directory Profiles and Searchbases” on page 488
- “Authentication and Authorization with Active Directory” on page 491
- “Configuring the credential_mapping Plug-In” on page 493
- “Configuring SSO for Use with Active Directory” on page 495
- “About Search Filters” on page 497
- “Configuring NetPoint for .NET Features” on page 497
- “Troubleshooting” on page 498
- “Microsoft Resources” on page 498

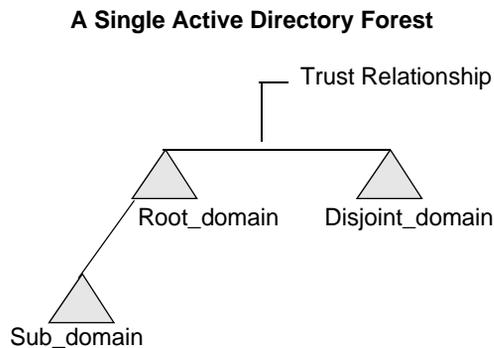
For additional information and procedures, see the *NetPoint 7.0 Installation Guide*. See also:

- “Configuring NetPoint for ADSI” on page 499
- “Configuring NetPoint for Active Directory with LDAP” on page 517

Setting Up Directory Profiles and Searchbases

The Active Directory forest shown below includes three domains: Root_domain, Sub_domain, and Disjoint_domain. The configuration in Figure 18 appears in the discussions that follow.

Figure 18 Three Domains in a Single Active Directory Forest



When you have finished NetPoint installation and set up for Active Directory, as described in the *NetPoint 7.0 Installation Guide*, you are ready to complete the following tasks.

- “Defining Directory Server Profiles for Remaining Domains” on page 488
- “Setting Up Disjoint Searchbases” on page 489
- “Configuring Group-Search Read Operations (Optional)” on page 490

Defining Directory Server Profiles for Remaining Domains

A default NetPoint directory server profile is created automatically each time you install the COREid Server and specify new directory server connection information. The directory server profile contains connection information for one or more directory servers that share the same namespace and operational requirements for read, write, search, and so on. The connection information includes a name, a domain or namespace to which it applies, a directory type, and a set of operations.

Note: The default NetPoint directory server profile is created for only your Root_domain. You must set up directory profiles for the remaining domains in your installation: for example, Disjoint_domain and Sub_domain.

After installation, you can use the COREid System Console to modify the directory server profiles as outlined below. When you finish the steps below, you may set up the Disjoint Searchbases.

For more information, see “Managing Directory Server Profiles” on page 290.

To set up additional directory server profiles

1. Navigate to the COREid System Console.
`http://hostname:port/identity/oblix`
2. Navigate to the directory server profile: CoreID System Console > System Admin > System Configuration > Configure Directory Options > *link*.
3. Add the profile for the Disjoint_domain, if you have one, as described in “Managing Directory Server Profiles” on page 290.
4. Add the profile for the Sub_domain.
5. Rename the Default Directory Profile *if* the default name generated during COREid System setup is not meaningful to you.
6. Set up disjoint searchbases, as described next.

Setting Up Disjoint Searchbases

After the domains are configured, you need to add a disjoint searchbase for the Disjoint_domain and verify that there is no value in the Tab Searchbase field.

Note: Depending on how you configured the Root_domain, you may want to add a disjoint searchbase for the Sub_Domain.

To add a disjoint searchbase for the Disjoint_domain

1. Navigate to the COREid System Console.
`http://hostname:port/identity/oblix`
2. Navigate to and select the Directory Server link: COREid System Console > System Admin > System Configuration > Configure Directory Options > *link*.
3. Add a disjoint searchbase for the Disjoint_domain and click Save.
4. Navigate to and select the Configure Tab function in the User Manager: COREid System Console > User Manager Configuration > Configure Tab.
5. Select a link on the Configure Tab page.
6. Confirm there is no value in the Tab Searchbase field.
7. Repeat the steps above for the Sub_domain, if you have one.

About Deleting a Disjoint Searchbase

If there are searchbase policies for the disjoint searchbase, a user who has this searchbase on this node is able to create a filter with Query Builder whose base is this searchbase. It is advisable to remove all access control policies for this disjoint searchbase before deleting it.

If you remove a disjoint searchbase, you must disable all the database agents that use this searchbase.

Configuring Group-Search Read Operations (Optional)

Active Directory uses incremental retrieval of group members. This means that NetPoint must perform multiple reads to get the complete set of group members.

For Active Directory on Windows 2000, the maximum number of members that can be retrieved with one read is 1000. Unless you change the parameter, NetPoint uses a default value of 1000. For Active Directory on Windows .NET Server 2003 the maximum is 1500. The parameter `maxForRangedMemberRetrieval` located in `globalparams.xml` contains the maximum value that NetPoint uses.

Note: The notation *install_dir* refers to the directory where you installed the named component. For example, *COREidServer_install_dir* refers to the directory where you installed the COREid Server.

To configure group-search read operations in NetPoint on Windows 2003

1. Locate the `globalparams.xml` file in
`\COREidServer_install_dir\identity\oblix\apps\common\bin\globalparams.xml`.
2. Add the `maxForRangedMemberRetrieval` entry with a value of 1500.

For example:

```
<SimpleList >  
<NameValPair ParamName="maxForRangedMemberRetrieval"  
Value="1500"/>  
</SimpleList>
```

3. Save the file.
4. Restart the COREid Server.

Authentication and Authorization with Active Directory

Two-forest configurations were introduced in the *NetPoint 7.0 Installation Guide*. After installation, configuring the Access System for Active Directory can include setting up authentication and authorization in a parent-child domain, as described below:

- “Parent-Child Authentication” on page 491
- “Parent-Child Authorization” on page 492
- “ObMyGroups Action Attribute” on page 492

Parent-Child Authentication

In this case, you need to use the NetPoint credential mapping plug-in to authenticate users against both the parent and child domains. This plug-in obtains the user’s DN.

For example, suppose you have two domains, `foo.goodwill.oblix.com` and `goodwill.oblix.com`. In the COREid System, you have two directory profiles, one for `foo.goodwill.oblix.com` with a display name of `foo` and another for `goodwill.oblix.com` with a display name of `goodwill`:

```
Foo.goodwill.oblix.com:  
DisplayName=foo  
Searchbase = dc=foo,dc=goodwill,dc=oblix,dc=com  
User:Alice
```

```
Goodwill.oblix.com  
DisplayName=goodwill  
Searchbase=dc=goodwill,dc=oblix,dc=com  
User:Bob
```

Also suppose you are using a Basic Over LDAP authentication mechanism that uses the filter in the `credential_mapping` plug-in shown above. When the user tries to log in to the Access System, a Basic Over LDAP dialog box appears.



The domain is the part of the login ID entered before the “\”. From the domain name, the Access System can tell which searchbase to use to identify this user. When Alice logs in, she must specify a domain name of foo, and when Bob logs in he must specify a domain name of goodwill. Both users will be authenticated.

Note: To access a resource that is protected by a Basic Over LDAP authentication scheme for an Active Directory forest, the user needs to enter the “*domainname\username*” as the user name in the Authentication dialog box. This *domainname* should be the display name for the DB profile created for the Access Server that is used to perform authentication for this user.

Parent-Child Authorization

You can define separate LDAP rules for each domain. For example, if you want all users who have the title of Manager to access a resource, you need to specify two LDAP rules, one for each domain. An example of these rules:

```
ldap:///dc=goodwill,dc=oblix,dc=com??sub?(&(title=Manager)
(objectclass=user)
```

```
ldap:///dc=foo,dc=goodwill,dc=oblix,dc=com??sub?(&
(title=Manager)(objectclass=user)
```

Using these rules, managers who are in both foo and goodwill can be authorized.

ObMyGroups Action Attribute

You can use the ObMyGroups action attribute to return in a header variable all of the groups to which a user belongs. Specifying ObMyGroups in the attribute name uses the searchbase configured for the Access System. Access Server does not impose any limit on the number of groups that are returned. The returned groups are only limited by any size limit configured for the directory.

The Access System supports only one searchbase. Therefore, if the user chooses goodwill.oblix.com as the product searchbase, then ObMyGroups results in a search for groups under goodwill.oblix.com. In this case, NetPoint cannot follow referrals, and since the Access System does not have multiple searchbase capability, groups from foo.goodwill.oblix.com cannot be returned.

You can specify ObMyGroups with an LDAP URL. In this case, the searchbase is picked up from the LDAP URL. However, you can only associate one attribute with one header variable, so if you have two domains you need at least two header variables to obtain all of the groups a user belongs to.

For example, suppose you have two domains: dc=goodwill,dc=oblix,dc=com and dc=dilbert,dc=goodwill,dc=oblix,dc=com. To obtain groups from both these searchbases you must define two separate header variables, one for each domain, as shown in Table 77.

Table 77 ObMyGroups with LDAP URLs for Two Domains

Type	Name	Return
headervar	HTTP_PARENT_GROUP	"obmygroups:ldap:///dc=goodwill,dc=oblix,dc=com??sub?(group_type=role)"
headervar	HTTP_CHILD_GROUP	"obmygroups:ldap:///dc=dilbert,dc=goodwill,dc=oblix,dc=com??sub?(group_type=role)"

- In HTTP_PARENT_GROUP: all the groups in "dc=goodwill,dc=oblix,dc=com" tree for which the logged-in user is a member and the group_type is role are returned.
- In HTTP_CHILD_GROUP: all the groups in "dc=dilbert,dc=goodwill,dc=oblix,dc=com" tree for which the logged-in user is a member and the group_type is role are returned.

The following procedures guide as you configure the credential_mapping authentication plug-in for Active Directory and set up SSO, if needed.

Configuring the credential_mapping Plug-In

Each policy domain requires an authentication scheme. Each authentication challenge method is supported by one or more plug-ins. For more information about NetPoint Plug-Ins for authentication challenge methods, see the *NetPoint 7.0 Administration Guide Volume 2*.

The `credential_mapping` plug-in maps the user's `userID` to a valid distinguished name (DN) in the directory. You can configure the attribute to which the `userID` is mapped. The most common attribute it is mapped to is `uid`. However, it is possible for a customer to map the `userID` to a profile attribute other than `uid` by changing the `obMappingFilter` parameter.

The `obmappingbase` defines the user searchbase. In a single domain, the mapping base must be explicitly defined in the `obMappingBase` parameter of the `credential_mapping` plug-in. For example, `ou=company,dc=mydomain,dc=oblix,dc=com`).

With an Active Directory forest, the user needs to provide the domain plus the user ID during the login to validate the user credential against the specified domain. In this case, the mapping base should be set to `obMappingBase="%domain%"`. The template for defining credential mapping for Basic over LDAP in an Active Directory forest should be `obmappingbase="%domain%",obmappingfilter=(amp(objectclass=user)(samaccountname=%userid%))", obdomain="domain"`

The domain information is maintained in the DB profiles in the COREid System Console. Be sure to create a DB profile for each domain. The login name for a multi-domain forest is the display name from the Access Server DB profile.

To configure the `credential_mapping` plug-in

1. Create a policy domain in the Access Manager, as usual.

Note: There is currently an upper limit of 350 policy domains when NetPoint is deployed with Active Directory.

2. Navigate to the Authentication Management Plug-In page: Access System Console > Access System Configuration > Authentication Management > *link* > Plug-Ins.

where *link* is the name of the authentication scheme you want to alter.

3. Configure the `credential_mapping` plug-in for Active Directory.

For example:

- For Form Based:

```
obmappingbase="%domain%",obmappingfilter=(amp(objectclass=user)(samaccountname=%login%))
```

- For Basic Over LDAP:

```
obmappingbase="%domain%",obmappingfilter=(&(objectclass=user)(sam
accountname=%userid%))", obdomain="domain"
```

Note: To access a resource that is protected by a Basic Over LDAP authentication scheme for an Active Directory forest, the user needs to enter the *domainname\username* as the user name in the Authentication dialog box. This *domainname* should be the display name for the DB profile created for the Access Server used to perform authentication for this user.

Configuring SSO for Use with Active Directory

You may want to configure SSO for the COREid or Access System, as described in the steps below. For more information, see about protecting resources with policy domains and configuring Single Sign-On, see the NetPoint 7.0 Administration Guide Volume 2.

To configure SSO with the COREid or the Access System

1. Change actions in the policy domain authorization rules that need to pass the header variable ObUniqueId (rather than uid) HTTP_OBLIX_UID.

Note: There is currently an upper limit of 350 policy domains with Active Directory. For details, see “Troubleshooting” on page 498.

2. On the Web server, modify the value of the WhichAttrIsLogin parameter to ObUniqueID in the files below.

For example:

```
\COREid_install_dir\identity\oblix\apps\common\bin\globalparams.xml
\AccessManager_install_dir\access\oblix\apps\common\bin\globalparams.lst
```

```
WhichAttrIsLogin:ObUniqueId
```

The following screens show examples of the associated directory profile and resulting basic login window related to the single sign-on.

This is the Configure Directory Server profiles page.

Configure Directory Server profiles
The following contains the Oblix Base and Searchbase settings. Click on the link to change a particular value.

Directory Server

Machine hobbes
Port number 389
Root DN cn=Administrator,cn=users,dc=dilbert,dc=goodwill,dc=oblix,dc=com
Root password <Not Displayed>
Search Base OU=Company,DC=goodwill,DC=oblix,DC=com
Oblix base ou=Oblix,ou=oblixree,dc=dilbert,dc=goodwill,dc=oblix,dc=com
Directory Server Security Mode Open
Disjoint Search Base
ADSI Enabled Yes

The following table contains the list of all Directory Profiles. Click on any link to change a particular profile. You must stop every OIS before the new values can take effect.

Name	Name Space	Primary Servers	Secondary Servers
<input type="checkbox"/> dilbert	ou=company,dc=dilbert,dc=goodwill,dc=oblix,dc=com	sagarwal2	
<input type="checkbox"/> default	ou=company,dc=goodwill,dc=oblix,dc=com	default	

This is the login dialog box.



If you are configuring for an Active Directory forest, the domain the user belongs to is determined by the directory profiles configured in the NetPoint COREid System and used by the Access System. These directory profiles can be enabled or disabled through the Configure Directory Server Profile page accessible from the COREid System Console, Configure Directory Options function.

- If a directory profile is disabled and the user enters that domain name during login through the Access System, then the user is not allowed access.
- If a directory profile is enabled and user enters that name as the domain name, then the user is allowed access.

However, if a user is already authenticated and has a valid session token, after which the directory profile is disabled, then the user is allowed access based on the authorization rules, and so forth. The directory profile state (enabled/disabled) does not take affect during authorization. Only authentication honors the state of the profile.

About Search Filters

Active Directory does not invoke indexed searches when the filter contains constraints that evaluate to *which contains*—filters such as `cn=*Smith`, where the searched for value contains *Smith*. For indexed searches the following constraints are valid:

- Equals (`cn=Smith`)
- Begins with (`cn=Smith*`)

The myGroups tab on the Group Manager may take a long time to evaluate the result if the dynamic groups options is enabled and the dynamic filters specified contain the *which contains* search filter.

To prevent users from using the *which contains* search filter, restrict the available filters by modifying the catalog files. In the following directory:

Component_install_dir/identity|access/oblix/app/application/bin

where *Component_install_dir* is the directory where the component is installed and identity|access represents either the COREid or Access system, respectively

There are several xxxparams.xml files. In these files you can specify the type of valid filters allowed under vallist - ObEnhanceSearchList.

See also, “Resolving Ambiguous Names” on page 523.

Configuring NetPoint for .NET Features

NetPoint provides support for .NET features with Windows Server 2003. For more information, see “Implementing .NET Features with NetPoint” on page 523.

For details about the following topics, see the *NetPoint Integration Guide*.

- Integrating NetPoint with the Authorization Manager
- Integrating NetPoint with Passport Authentication
- Integrating NetPoint with Smart Card Authentication
- Integrating NetPoint with the Security Connector for ASP.NET

Troubleshooting

Active Directory Search Halts

Symptom—400 policy domains were created in NetPoint, each with 10 resources and 10 policies. The `limitAMPolicyDomainResourceDisplay` is set to true in the Access Manager `globalparams.lst` file. When the Search icon is selected, an error page appears stating, “The following messages were produced by the product. Please contact your webmaster to fix the problem.”

Cause—The number of policy domains exceeds the current limit.

Solution—Do not exceed 350 policy domains with Active Directory.

Adding Members to Static Groups Causes the Group Size to Shrink

Symptom—Adding users to static groups works properly up to a point, after which continuing to add members causes the group size to shrink.

Solution—Change the value for the parameter `maxForRangedMemberRetrieval` in `globalparams.xml` to a number higher than the desired group membership size. If you are using Active Directory on Windows 2003, set the parameter `maxForRangedMemberRetrieval` in `globalparams.xml` to 1500. If you are using Active Directory on Windows 2000, set it to 1000.

Microsoft Resources

Active Directory Home Page

<http://www.microsoft.com/windows2000/technologies/directory/ad/default.asp>

ADSI Overview

<http://www.microsoft.com/windows2000/techinfo/howitworks/activedirectory/adsilinks.asp>

Active Directory Programmers Page

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netdir/adsi/active_directory_service_interfaces_adsi.asp?frame=true

ADSI Programmers Page

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netdir/adsi/active_directory_service_interfaces_adsi.asp?frame=true

B Configuring NetPoint for ADSI

Both the NetPoint COREid System and the NetPoint Access System provide support for Active Directory Services Interface (ADSI) client applications. This chapter summarizes requirements and procedures when you are running NetPoint with Active Directory forests and the Active Directory Services Interface (ADSI). See:

- “About ADSI with NetPoint” on page 499
- “COREid System ADSI Configurations” on page 501
- “Access System ADSI Configurations” on page 506
- “Configuring ADSI for the COREid System” on page 510
- “Enabling ADSI for a Default Directory Profile” on page 510
- “Enabling ADSI for Other Directory Profiles” on page 511
- “Configuring ADSI for the Access System” on page 513
- “Changing the pageSize Parameter” on page 515

For additional information and procedures, see the *NetPoint 7.0 Installation Guide*.

About ADSI with NetPoint

Active Directory runs on Windows® 2000 and Windows Server 2003 domain controllers. Client applications using ADSI may be written and run on other windows platforms.

ADSI is a set of COM interfaces that enable tight integration with Active Directory. For example, ADSI:

- Abstracts the capabilities of different directory services from multiple vendors to present a single interface for managing network resources.
- Allows administrators and developers to manage the resources in a directory service, regardless of which network environment contains the resource.

- Enables administrators to automate common tasks such as adding users and groups, managing printers, and setting permissions on network resources.

Important: Enabling ADSI allows NetPoint to take advantage of Active Directory's implicit failover and password-change capabilities.

With ADSI, NetPoint components do not have to bind to a specific host and port to access Active Directory data. Instead, ADSI allows NetPoint components to connect to the nearest available domain controller for accessing any user, group, or NetPoint configuration information.

As described in the *NetPoint 7.0 Installation Guide*, the credentials for ADSI are used to bind to the entire forest. A forest can contain multiple Active Directory hosts. When user data and configuration data are stored on separate Active Directory hosts in separate forests, you cannot connect to these simultaneously using ADSI.

ADSI does *not* require specific host and port numbers for different domains in the forest. ADSI connects to Active Directory hosts using an LDAP URL like the one shown below:

```
LDAP://domain.oblix.com/ou=oblix,dc=domain,dc=oblix,dc=com
```

For details about enabling ADSI during NetPoint installation, see the *NetPoint 7.0 Installation Guide*.

Recommendation

Active Directory replicates the entire tree structure. Due to potential replication delays, Oblix recommends you not replicate the directory tree containing Oblix data. Changes to Oblix data may not be immediately available. For example, a change made to a user's access permissions in the NetPoint Access Manager may not be available to the Access Server if they are talking to different domain controllers.

If you must replicate the Oblix tree, modify the replication frequency between the domain controllers on Active Directory.

COREid System ADSI Configurations

NetPoint supports a flexible combination of ADSI and LDAP that relates to your choice of authentication options and binding options.

Note: SSL is not *required* with ADSI and NetPoint. However, your business may require SSL for other reasons. For example, directory binds are in clear text and SSL is not automatically provided.

Discussions here include the following:

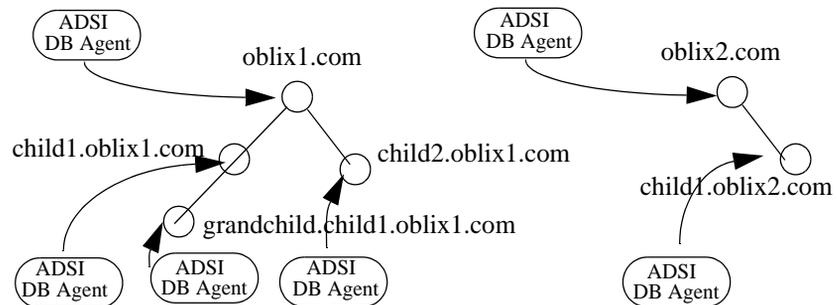
- “Pure ADSI with ADSI Authentication” on page 501
- “Mixed ADSI with LDAP Authentication” on page 502
- “NetPoint COREid ADSI Configuration Files” on page 503

Pure ADSI with ADSI Authentication

With a *pure* ADSI setup, a single ADSI database agent is created during COREid System setup for the primary domain controller in your Active Directory tree. You must add additional agents for each child domain.

Additionally, if you have a noncontiguous forest of domain trees, you need to associate a separate ADSI database agent for each primary domain controller as shown in Figure 19.

Figure 19 Noncontiguous Forest with ADSI



See “Managing Directory Server Profiles” on page 290 for more information about multiple directory profiles and DB agents.

Mixed ADSI with LDAP Authentication

ADSI authentication may be slower than LDAP. For this reason, you may want to use LDAP for authentication when other operations, such as read, write, and search are handled by ADSI. An ADSI agent must be associated with every domain.

To associate an ADSI agent with every domain

1. Select the “Use LDAP for authentication” check box next to Microsoft Active Directory (using ADSI) on the Create Directory Profile page in the NetPoint COREid System Console.

Directory Type	<input type="radio"/> IBM Secure Way Directory
	<input checked="" type="radio"/> Microsoft Active Directory (using ADSI)
	<input checked="" type="checkbox"/> Use LDAP for Authentication
	<input type="radio"/> Microsoft Active Directory

See “Managing Directory Server Profiles” on page 290 for more information about multiple directory profiles and DB agents.

2. Repeat as needed.

Note: The Global Catalog was required with earlier releases of NetPoint and is no longer required with NetPoint v6.5 and later.

For more information, see “Bind Mechanisms for the Access Server” on page 507 and “NetPoint COREid ADSI Configuration Files” on page 503.

Bind Mechanisms for the COREid Server

ADSI provides several ways for the COREid Server to bind to Active Directory. There is no advantage to any particular method. Instead, it depends on which credentials you want to use. For example:

- **Implicitly**—Using the credential of the current process. This is the default for the COREid Server.

This corresponds to the service logon credentials for the COREid Server. For implicit bind, the useImplicitBind flag in the adsi_params.xml file should be set to 0. See “NetPoint COREid ADSI Configuration Files” on page 503.

Note: You must create an account for the COREid Server to bind to Active Directory.

The account that enables the COREid Server to bind to the Active Directory must be equivalent to the root DN that you specify during setup of the COREid Server. It should have all the administrative privileges for operations that are to be performed using NetPoint. In an Active Directory forest, this user should be delegated control over all the other domains in the forest.

- **Explicitly Using the DN of the user**—The useImplicitBind flag in the adsi_params.xml file should be set to 1. The user DN should be specified with the adsiCredential parameter located in the adsi_params.xml file. See “NetPoint COREid ADSI Configuration Files” on page 503.
- **Explicitly Using the userPrincipleName**—The useImplicitBind flag in the adsi_params.xml file should be set to 2. The UPN should be specified in the adsiUPN parameter in the adsi_para.xml file. See “NetPoint COREid ADSI Configuration Files” on page 503.

NetPoint COREid ADSI Configuration Files

ADSI configuration parameters are maintained in two files:

```
\COREid_install_dir\identity\oblix\apps\common\bin\globalparams.xml
\COREid_install_dir\identity\oblix\config\adsi_params.xml
```

where *COREid_install_dir* is the directory in which you installed the COREid Server.

About globalparams

This section shows a sample globalparams.xml file followed by a table of parameter values.

The install program in *\COREid_install_dir\identity\oblix\apps\common\bin\globalparams.xml* creates the adsiEnable parameter and sets its value to true when you enable ADSI for the default directory profile. This parameter refers to a system level directory profile that contains Oblix data.

Note: You must restart the COREid Server after changing any parameters. However, do not change the ADSIEnabled parameter value.

Parameters and values for globalparams are shown in Table 78.

```
<SimpleList>
<NameValPair ParamName="ActiveDirectory" Value="true" />
</SimpleList>
<SimpleList>
<NameValPair ParamName="ADSIEnabled" Value="true" />
</SimpleList>
```

Table 78 Parameters and Values in globalparams Files

globalparams Parameters	Values
ActiveDirectory	true false True when the NetPoint Administrator selects Active Directory as the directory server type during COREid Server configuration
ADSIEnabled	true false True when the NetPoint Administrator enables ADSI during COREid Server configuration

About adsi_params

This section shows a sample adsi_params.xml file followed by a table of parameter values. By default, adsi_params.xml includes a value for the adsiCredential parameter and the password, as shown below. This enables you to change the bind mechanism to be Explicit after initial setup.

The adsiPassword is encrypted and can only be generated by NetPoint during setup. An example of this file in NetPoint is shown below:

```
<?xml version="1.0" ?>
- <ParamsCtI g xml ns="http://www.obl i x. com" CtI gName="adsI _params">
  - <CompoundLi st Li stName="adsI _params">
    - <Val NameLi st Li stName="adsI _params">
      <NameVal Pai r ParamName="si zeLi mi t" Val ue="0" />
      <NameVal Pai r ParamName="ti meLi mi t" Val ue="0" />
      <NameVal Pai r ParamName="pagesi ze" Val ue="100" />
      <NameVal Pai r ParamName="useI mpl i ci tBi nd" Val ue="0" />
      <NameVal Pai r ParamName="adsI Credi ti al "
Val ue="cn=Admi ni strator, cn=users, dc=goodwi l l , dc=obl i x, dc=com" />
      <NameVal Pai r ParamName="adsI Password" Val ue="0243455B5B5F5C4C5651595D41"
/>
    />
      <NameVal Pai r ParamName="useGCForAuthn" Val ue="fal se" />
      <NameVal Pai r ParamName="encrypti on" Val ue="fal se" />
      <NameVal Pai r ParamName="asynchronousSearch" Val ue="true" />
      <NameVal Pai r ParamName="useDNSPrefi xedLDAPPaths" Val ue="fal se" />
    </Val NameLi st>
  </CompoundLi st>
</ParamsCtI g>
```

By default, encryption is set to false in `adsi_params.xml`. If you set it to true when running in open mode and restart the COREid Server, the COREid Server will not work.

Note: You must restart the COREid Server after changing any parameters.

Table 79 describes parameters and values within the `adsi_params` files, next.

Table 79 Parameters and Values in `adsi_params` Files

adsi_params Parameters	Values
sizeLimit	Integer value that limits the number of query results returned for authentication.
timeLimit	Integer value that limits the number of seconds before a query times out.
pageSize	Page size of results that ADSI request from the server.
useImplicitBind	0 = Implicit credentials 1 = Explicit credentials 2 = Use userPrincipalName
adsiCredential	An LDAP specification of a user, such as <code>cn=Administrator,cn=users,dc=myhost,dc=mydomain,dc=com</code>
adsiPassword	An encoded text string representing the LDAP user's password.
useGCForAuthn	true/false False
asynchronousSearch	true/false By default ADSI is enabled to perform asynchronous searches. If set to false, it does synchronous searches.
adsiUPN	This parameter needs to be added if <code>useImplicitBind</code> is set to 2. The value of the parameter should be the UPN (<code>userPrincipalName</code>) of the user.
pageSize	Setting the <code>pageSize</code> value to a finite value (the default is 0) turns off LDAP referrals. This can improve performance when client applications perform directory searches.
chaseReferral	Setting this flag to false turns off LDAP referrals.

Access System ADSI Configurations

Like the COREid System, the NetPoint Access System provides the support for both ADSI and ADSI with LDAP authentication.

Unlike the COREid System, the NetPoint Access System does not include the user interface for adding multiple directory profiles. However, the Access System does support multiple Active Directory domains and you must perform the steps for enabling ADSI for the default directory profile created during COREid setup.

Discussions here include the following:

- “Pure ADSI with ADSI Authentication” on page 507
- “Access System ADSI Configuration Files” on page 508

Pure ADSI with ADSI Authentication

The Access Server authenticates to Active Directory using ADSI. This is the default when you enable ADSI on these components.

- The Access Manager uses the same authentication mode as the COREid Server. Still, you must enable ADSI for the Access Manager. For more information, see “Configuring ADSI for the Access System” on page 513.
- The Access Server can communicate directly with all directory servers in the forest and no longer requires the Global Catalog for LDAP authentication.

For a list of ADSI installation and setup considerations, see the *NetPoint 7.0 Installation Guide* appendix, “Installing NetPoint with Active Directory.”

Authentication Mechanisms

When users authenticate to Active Directory, the mechanism is the domain controller, which uses respective domain controllers for authentication with ADSI.

See “Access System ADSI Configuration Files” on page 508 for more information.

Bind Mechanisms for the Access Server

ADSI provides several ways for the Access Server and Access Manager to bind to Active Directory. There is no advantage to a particular method. Instead, it depends on which credentials you wish to use:

- **Implicitly**—Using the credential of the current process (default for the Access Server). This corresponds to the service logon credentials for the Access Server. For implicit bind, the `useImplicitBind` flag in the `adsi_params.lst` file should be set to 0. See “Access System ADSI Configuration Files” on page 508.

Note: You need to create an account for the Access Service to bind to Active Directory. This account must be equivalent to the Root DN that you specify during setup of the Access Server. It should have all the administrative privileges for the operations that are to be performed using NetPoint. In an Active Directory Forest, this user should be delegated control over all the other domains in the forest.

- **Explicitly Using the DN of the User**—The `useImplicitBind` flag in the `adsi_params.lst` file should be set to 1. The user DN should be specified with

the `adsiCredential` parameter located in the `adsi_params.lst` file. See “Access System ADSI Configuration Files” on page 508.

- **Explicitly Using the `userPrincipalName`**—The `useImplicitBind` flag in the `adsi_params.lst` file should be set to 2. The UPN should be specified in the `adsiUPN` parameter in the `adsi_params.lst` file. See “Access System ADSI Configuration Files” on page 508 for more information.

In a multi-domain Active Directory forest the only supported explicit bind mechanism is `userPrincipalName`. The Access Manager supports only this mechanism.

Access System ADSI Configuration Files

Both the Access Manager and Access Server each have two configuration files for modifying ADSI related parameters. Although the files are maintained in separate locations, and must be modified separately for each component, their contents are the same. The configuration files for the Access Manager and Access Server are listed below:

```
\AccessManager_install_dir\access\oblix\apps\common\bin\globalparams.lst
\AccessManager_install_dir\access\oblix\config\adsi_params.lst

\AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.lst
\AccessServer_install_dir\access\oblix\config\adsi_params.lst
```

Access Manager ADSI Configuration

This section shows a sample `global-parameters` configuration file, followed by a table of parameter values.

Note: When you install Access Manager and Access Server, if you do not choose the ADSI option, you do *not* see the `ADSIEnabled` parameter in `globalparams.lst`. However, you do still see the `useLDAPBind` parameter, though it serves no purpose without `ADSIEnabled`.

```
BEGIN: vCompoundList
...
useLDAPBind: false
ADSIEnabled: true
ActiveDirectory: true
END: vCompoundList
```

The parameters and their values are described in Table 80.

Table 80 Parameters and Values in globalparams Files

globalparams Parameters	Values
useLDAPBind	true false True when the NetPoint Administrator selects “Microsoft Active Directory using LDAP” during Access Manager configuration. The ADSIEnabled flag must be true for this flag to have effect. The default is false.
ADSIEnabled	true false True when the NetPoint Administrator s enables ADSI during Access Manager configuration.
ActiveDirectory	true false True when the NetPoint Administrator selects Active Directory as the Directory Server type during Access Manager configuration.

Access Server ADSI Configuration

This section shows a sample adsi parameters configuration file, followed by Table 81 of parameter values.

Table 81 Parameters and Values in adsi_params Files

adsi_params Parameters	Values
sizeLimit	Integer value that limits the number of query results returned for authentication.
timeLimit	Integer value that limits the number of seconds before a query times out.
pageSize	Page size of results that adsi request from the server. The default is 0.
useImplicitBind	0 = Implicit Credentials 1 = Explicit Credentials 2 = Use UserPrincipalName
adsiCredential	An LDAP specification of a user, such as “cn=Administrator,cn=users,dc=myhost,dc=mydomain,dc=com”
adsiPassword	An encoded text string representing the LDAP user's password.
adsiUPN	Text string of UserPrincipalName when use ImplicitBind=2. A UPN string is typically an email address with the format: user@company.com
useGCForAuthn	True/False Change the useGCForAuthentication parameter to false.

Table 81 Parameters and Values in adsi_params Files

ads_i_params Parameters	Values
asynchronousSearch	True/False By default, ADSI is enabled to perform asynchronous searches. If set to false, it does synchronous searches
ads_iUPN	This parameter needs to be added if useImplicitBind is set to 2. The value of the parameter should be the UPN (userPrincipalName) of the user.

Configuring ADSI for the COREid System

There are several tasks involved in configuring ADSI for the COREid System. Details are provided in the *NetPoint 7.0 Installation Guide*.

Task overview: Configuring ADSI for the COREid System

1. Prepare your Active Directory, as described in your Microsoft documentation and the *NetPoint 7.0 Installation Guide*.
2. Specify ADSI when you install and set up NetPoint, as described in the *NetPoint 7.0 Installation Guide*.

By default, this creates a pure ADSI configuration in which a single ADSI directory profile (DB agent) enables associated COREid Servers to perform all operations with a primary domain controller in your Active Directory tree using an Implicit Bind.

3. Set up Active Directory attributes and/or enable change-password permissions, as described in the *NetPoint 7.0 Installation Guide*.
4. Configure the default directory profile, as described in “Enabling ADSI for a Default Directory Profile” on page 510.
5. Enable ADSI for additional directory profiles, if desired, as discussed under “Enabling ADSI for Other Directory Profiles” on page 511.

Enabling ADSI for a Default Directory Profile

The NetPoint COREid System automatically creates a default directory profile during installation. You can enable ADSI for the default profile during COREid System setup.

The default database agent is automatically assigned a name using the convention *default-ois-machine name*. You should modify this name to the respective domain name because users must enter this name during authentication.

Enabling ADSI for Other Directory Profiles

If you have a noncontiguous forest of domain trees, you need to associate a separate ADSI database agent for each primary domain controller. Additional directory profiles are configured after COREid System installation and set up, as outlined below and described in the “COREid System Administration” chapter.

To enable ADSI for additional directory profiles

1. Navigate to the COREid System Console.
`http://hostname:port/identity/oblix`
2. Navigate to Configure Directory Options page: COREid System Console > System Admin > System Configuration > Configure Directory Options.
3. Click the Add button to display the Create Directory Server profile page.

The screenshot shows the 'Create Directory Server profile' page in the COREid System Console. The page has a navigation menu on the left with options like 'Configure Admins', 'Configure Styles', 'Import Photos', 'View Server Settings', 'Configure Directory Options' (selected), 'Configure Webpass', 'Configure Password Policy', and 'Configure COREid Server'. The main content area includes the following fields and options:

- Name***: Text input field.
- Name Space***: Text input field.
- Directory Type**: Radio button options: iPlanet Directory Server 4.x, iPlanet Directory Server 5.x, Novell Directory Services (NDS eDirectory), IBM Secure Way Directory, Microsoft Active Directory (using ADSI) (selected), and Microsoft Active Directory. Under 'Microsoft Active Directory (using ADSI)', there is a checkbox for 'Use LDAP for Authentication'.
- Dynamic Auxiliary**: Radio button options: Yes (selected), No.
- Operations**: Radio button options: All Operations (selected), Selected Operations. Under 'Selected Operations', there are three sections: Search (Search Entries checked, Authenticate User checked), Read (Read Entry checked), and Write (Create Entry checked, Delete Entry checked, Modify Entry checked, Change Password checked).
- Used By**: Radio button options: All NetPoint Components, COREid servers. A dropdown menu is open showing 'All servers' and '65fcs11a-COREid1-yttrium-6500'.

Oblix recommends using the respective domain names as the profile names because users must enter this name during authentication.

4. Enter a name for this directory profile.

You must configure a directory profile for each Domain and Sub_domain controller. For more information, see “Configuring ADSI for the COREid System” on page 510.

5. Enter a namespace for this directory profile.

There are multiple choices for the directory type. To use Active Directory without ADSI enabled, you should select Microsoft Active Directory.

Note: You still have the option to enable ADSI or SSL for changing passwords. Also, you can enable LDAP by selecting the secondary check box, Use LDAP for Authentication. When LDAP is enabled, an ADSI DB Agent associates with the primary domain controller. An LDAP agent needs to be configured for whichever Sub_domain controller you want to use to authenticate.

6. Select the appropriate directory type. For example:

Directory Type	<input type="radio"/> LDAP Secure way Directory
	<input type="radio"/> Microsoft Active Directory (using ADSI) <input type="checkbox"/> Use LDAP for Authentication
	<input checked="" type="radio"/> Microsoft Active Directory
	AD-Change password using: <input checked="" type="radio"/> ADSI <input type="radio"/> SSL

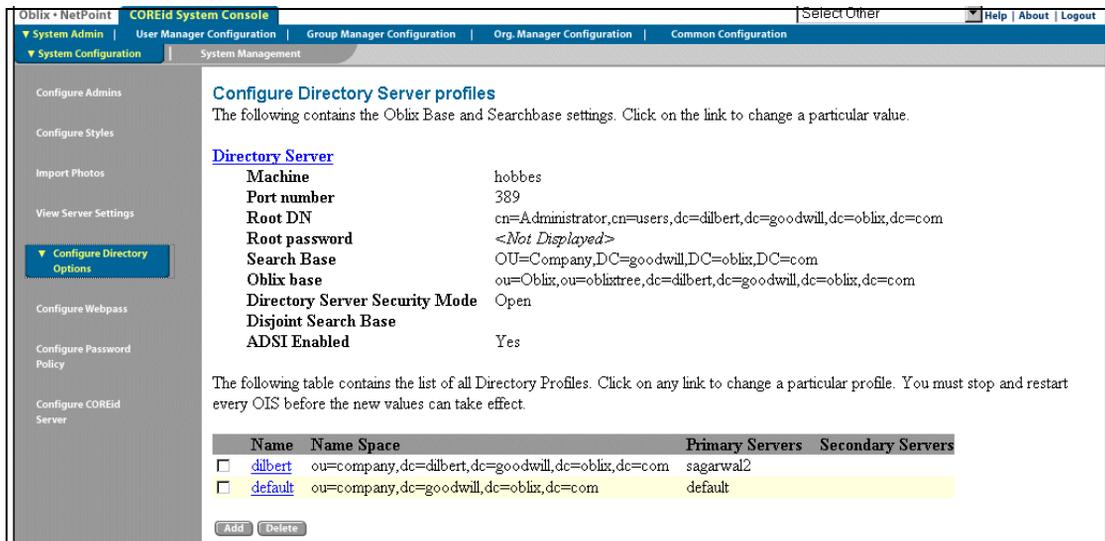
7. Select the operations supported for this directory profile, for example:

<input checked="" type="radio"/> All Operations												
<input type="radio"/> Selected Operations												
<table border="1"><tr><td>Search</td><td><input checked="" type="checkbox"/> Search Entries</td><td><input checked="" type="checkbox"/> Authenticate User</td></tr><tr><td>Read</td><td><input checked="" type="checkbox"/> Read Entry</td><td></td></tr><tr><td>Write</td><td><input checked="" type="checkbox"/> Create Entry</td><td><input checked="" type="checkbox"/> Modify Entry</td></tr><tr><td></td><td><input checked="" type="checkbox"/> Delete Entry</td><td><input checked="" type="checkbox"/> Change Password</td></tr></table>	Search	<input checked="" type="checkbox"/> Search Entries	<input checked="" type="checkbox"/> Authenticate User	Read	<input checked="" type="checkbox"/> Read Entry		Write	<input checked="" type="checkbox"/> Create Entry	<input checked="" type="checkbox"/> Modify Entry		<input checked="" type="checkbox"/> Delete Entry	<input checked="" type="checkbox"/> Change Password
Search	<input checked="" type="checkbox"/> Search Entries	<input checked="" type="checkbox"/> Authenticate User										
Read	<input checked="" type="checkbox"/> Read Entry											
Write	<input checked="" type="checkbox"/> Create Entry	<input checked="" type="checkbox"/> Modify Entry										
	<input checked="" type="checkbox"/> Delete Entry	<input checked="" type="checkbox"/> Change Password										

If this is a directory profile configured for a domain controller, select all operations.

8. Complete the rest of the directory profile and save it, as usual.

See the following screen for a completed directory profile page. See also “Managing Directory Server Profiles” on page 290 for more information about multiple directory profiles (DB agents).



Configuring ADSI for the Access System

The Access Manager uses the COREid Server for authentication. Therefore, the login operation uses the same mode (ADSI or LDAP) as the COREid Server it talks to. During Access Manager setup, by default, you use an Explicit Bind to enable the Access Manager and Access System Console to perform all operations except authentication in the Active Directory tree.

Note: SSL is *not required* for ADSI configurations with NetPoint. However your business may require SSL for other reasons. For example, directory binds are in clear text, and SSL is not automatically provided.

By default, enabling ADSI for the Access Server creates a pure ADSI configuration in which the Access Server performs all operations with a primary domain controller in your Active Directory tree using an Implicit Bind.

Configuring ADSI support in the NetPoint Access System involves the tasks below.

Task overview: Configure ADSI for the Access System

1. Validate your COREid setup, as described in the *NetPoint 7.0 Installation Guide* appendix.
2. Install and set up the Access Manager, as described in the *NetPoint 7.0 Installation Guide*.
3. Install the Access Server and setup ADSI, as described in the *NetPoint 7.0 Installation Guide*.

4. Install the WebGate, as described in the *NetPoint 7.0 Installation Guide*.
5. Enable LDAP authentication for the Access Server, if desired, as described in “Enabling LDAP Authentication for the Access Server” on page 514

Enabling LDAP Authentication for the Access Server

ADSI authentication may be slower than LDAP. For that reason, you may wish to use LDAP for authentication while other operations such as authorization and auditing are handled by ADSI.

To enable LDAP authentication for the Access Server

1. Open
`\AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.lst`
with a text editor.
2. Change the value of useLDAPBind to true.
3. Save globalparams.lst.
4. Create a copy of ConfigDBfailover.lst located in
`\AccessServer_install_dir\access\oblix\apps\config` and name it
AppDBfailover.lst.

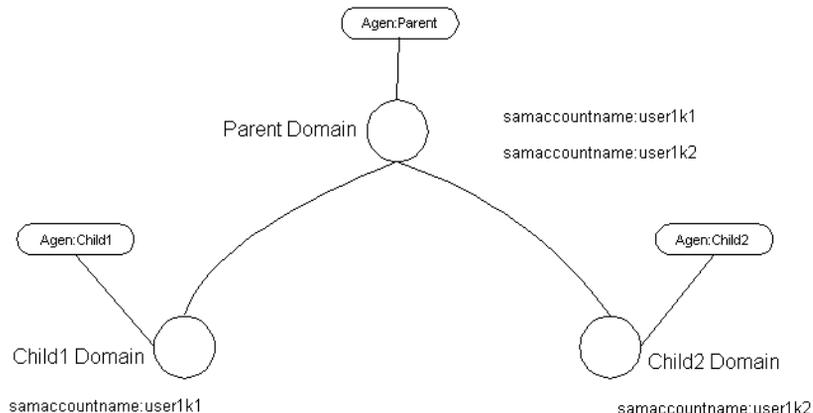
Both files should reside in the same directory.

5. Save.
6. Restart the Access Server.

Changing the pageSize Parameter

Based on your Active Directory forest deployment you may need to change the page-size parameter in the adsi_params file. For example, in Figure 20 you have a parent-child relationship between your Active Directory domains, and you have user(s) in both the parent and child domain with the same samaccountname.

Figure 20 Users in Both the Parent and Child Domains



Assume the authentication mechanism is Basic over LDAP for an Active Directory forest. In this case:

- For user1k1 to log into the Child1 domain, the user can enter their userid as Child1\user1k1.
- For user1k2 to log into the Child2 domain, the user can enter their userid as Child2\user1k2.

However, if the pageSize parameter is set to 0, for user1k1 from the parent domain to log in entering Parent\user1k1 produces an error: “The credentials Parent\user1k1 used in the login correspond to more than one user profile in the Identity System. The correspondence must be unique.”

This is because when the page size is set to 0, ADSI searches the subdomains, therefore finding two users who satisfy the criteria. For user1k1 and user1k2 to log into the parent domain you need to set the pageSize parameter to a finite value. Oblix recommends using 100.

Troubleshooting

ADSI Cannot Be Enabled for this DB Profile

NetPoint supports changing the user data DB profile between ADSI and LDAP using the NetPoint System Console. However, NetPoint does not support changing the Oblix and/or policy DB profile between ADSI and LDAP using the NetPoint System Console.

Symptom—Suppose you have user data stored in an Active Directory forest using LDAP and NetPoint configuration and policy data stored in another Active Directory forest using ADSI. When you change the ADSI flag in the Oblix data DB profile to LDAP—using the NetPoint System Console—and restart NetPoint servers and services, the ADSI flag remains enabled, and the message below appears:

“ADSI can be enabled for either User or Oblix DB Profile if they are in a separate forest. ADSI Cannot be Enabled for this DB Profile.”

Further, attempting to modify the user data DB profile to ADSI produces an error, because NetPoint recognizes the DB profile for Oblix and policy data as ADSI enabled.

Solution—Rerun the setup program to modify the DB profile for Oblix configuration and policy data.

C Configuring NetPoint for Active Directory with LDAP

This chapter summarizes procedures to set up NetPoint with Active Directory forests using LDAP as the communication protocol. See:

- “Overview” on page 517
- “Setting Up the Access Manager for LDAP” on page 518
- “Setting Up the Access Server for LDAP” on page 519
- “Setting Active Directory Timeouts for LDAP” on page 520
- “Enabling LDAP Authentication with ADSI” on page 521

For additional information and procedures, see the *NetPoint 7.0 Installation Guide*.

Note: The instructions here apply only if you are using LDAP as the protocol between the Access System and Active Directory. If your environment differs, skip this discussion.

Overview

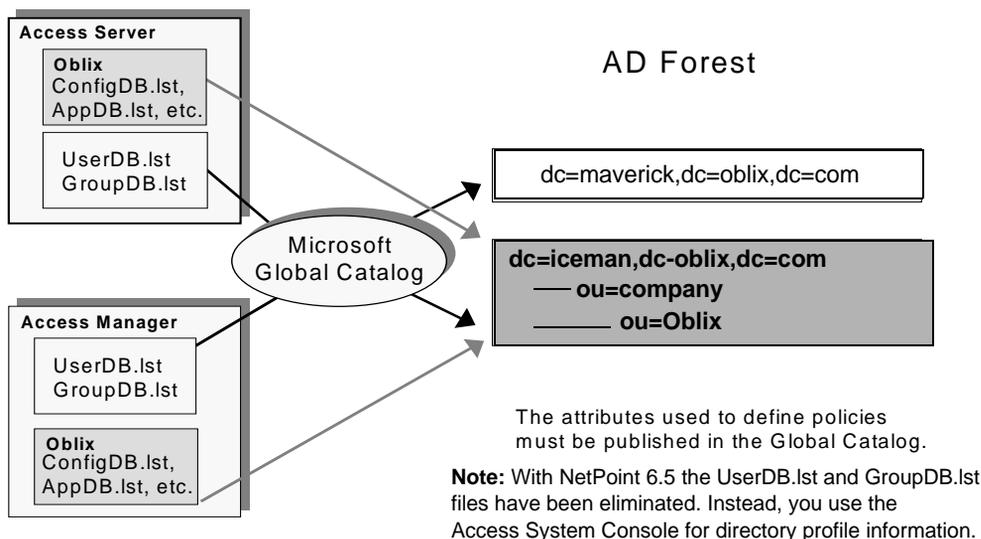
The Access System supports Active Directory forests with some modifications.

Note: The Microsoft Global Catalog is no longer required for the Access System.

The steps in this section are based on the example below. In this case, the COREid System was configured using the two domains shown in Figure 21. For example:

- dc=maverick,dc=oblix,dc=com
- dc=iceman,dc=oblix,dc=com

Figure 21 Active Directory Forest with Two Domains



Complete the procedures below to set up the Access System for multiple domains using LDAP between the Access System and Active Directory:

- “Setting Up the Access Manager for LDAP” on page 518
- “Setting Up the Access Server for LDAP” on page 519
- “Setting Active Directory Timeouts for LDAP” on page 520

Note: In the discussions below, *_install_dir* refers to the installation directory you specified for the named component. For example, *AccessManager_install_dir* is the directory where you installed the Access Manager.

Setting Up the Access Manager for LDAP

The Oblix-related configuration information, listed below, is located in the `\AccessManager_install_dir\access\oblix\config\ldap` directory and must be accessed directly:

- AppDB.lst
- ConfigDB.lst
- WebResrcDB.lst

Suppose the COREid System was set up as shown earlier with the Oblix data in the `dc=iceman,dc=oblix,dc=com` domain.

In this case, the Access Manager must also be set up for this domain. To accomplish this, you must specify the same configuration DN for the Access Manager and the COREid Server.

For more information about the configuration DN, see the *NetPoint 7.0 Installation Guide*.

To set up the Access Manager for Active Directory

1. Navigate to the Access Manager setup page:

`http://hostname:port/access/oblix`

2. Set up the Access Manager with the same configuration DN as the COREid Server.

For example, against the machine containing the domain:

`dc=iceman,dc=oblix,dc=com`

3. Before starting the Web server, change the port to 3268 (open LDAP) to ensure that users and groups are accessible from both domains.
4. See the appendix “Deploying NetPoint with Active Directory” for information about configuring the `credential_mapping` plug-in required in your authentication schemes and setting up SSO.

Setting Up the Access Server for LDAP

This section only applies if you are using LDAP as the protocol between the Access Server and Active Directory.

To set up the Access Server for Active Directory

1. Configure the Access Server using the same configuration DN as the COREid Server.

For example, against the machine containing:

`dc=iceman,dc=oblix,dc=com domain`

2. Make sure Active Directory time out is handled correctly, as described under “Setting Active Directory Timeouts for LDAP” on page 520.
3. Create a copy of `ConfigDBfailover.lst` located in `\AccessServer_install_dir\access\oblix\apps\config` and name it `AppDBfailover.lst`.

Both files should reside in the same directory.

Setting Active Directory Timeouts for LDAP

If you are using LDAP, you need to configure timeouts for the Access Server when it is installed against Active Directory.

The Access Server, which runs as a service, opens connections to Active Directory. Active Directory times out idle connections after a period of inactivity, which means that the Access Server can try to access the directory and fail.

If you want to avoid this problem, you need to establish new connections before the Active Directory Idle Session Time is reached. The failover information can be specified when:

- You are prompted to specify failover information at the end of the Access Server installation.
- You reconfigure failover information after installation is complete, using the `configureAAAServer` application in `AccessServer_install_dir/access/oblix/tools/configureAAAServer`, as described below.

The following files are created when you use the `ConfigAAAServer.exe` tool to configure failover between a second directory server and the Access Server:

`ConfigDBfailover`
`AppDBfailover`
`Web...DBfailover`

To specify Access Server failover after installation

1. Locate the `configureAAAServer` application.
`\AccessServer_install_dir\access\oblix\tools\configureAAAServer`
2. Launch the `configureAAAServer` application using the command below.
`configureAAAServer install AS_install_dir`
3. Answer No, when asked if you want to reconfigure Access Server.
4. Answer Yes, when asked if you want to specify failover information.
5. When asked where different types of data are stored, respond appropriately for your environment.

For example:

- **Separate Directory Servers, Choose Option 8**—If policy and Configuration DN are separate from user data, choose option 8 (Modify Common Parameters) and specify values appropriate for your system.
- **Same Directory Server, Choose Option 4**—If policy, Configuration DN, and user information is on the same directory server, choose option 4

(Modify Common Parameters) and enter values appropriate for your system.
For example:

Maximum Connections: 1
Sleep For (seconds): 60
Failover Threshold: 1
Maximum Session Time (seconds): 120

After every Maximum Session Time, a new connection is created to Active Directory, and the old connection is dropped, whether the Access Server was idle or not.

Note: Make sure the Maximum Session Time (in seconds) is less than the Active Directory Idle Timeout (typically less than 600 seconds).

6. Choose the option to quit.
7. When asked if you want to commit the changes, answer Yes.

For more information about failover, see the *NetPoint 7.0 Deployment Guide*.

Enabling LDAP Authentication with ADSI

ADSI authentication may be slower than LDAP. For that reason, you may wish to use LDAP for authentication while other operations such as authorization and auditing are handled by ADSI.

To enable LDAP authentication for the Access Server

1. Open `globalparams.lst` with a text editor.
`\AccessServer_install_dir\access\oblix\apps\common\bin\globalparams.lst`
2. Change the value of `useLDAPBind` to `true`.
3. Save `globalparams.lst`.
4. Create a copy of `ConfigDBfailover.lst` located in:
`\AccessServer_install_dir\access\oblix\config\ldap\ConfigDBfailover.lst`
5. Name it `AppDBfailover.lst`.
Both files should reside in the same directory.
6. Save.

D Implementing .NET Features with NetPoint

NetPoint provides support for .NET features with Windows Server 2003. For details about supported features and their implementation within NetPoint, see the following topics in this chapter:

- “Resolving Ambiguous Names” on page 523
- “Configuring NetPoint for Dynamically Linked Auxiliary Classes” on page 529
- “Enabling Fast Bind for NetPoint Authentication” on page 534
- “Enabling Impersonation” on page 536
- “Setting Up Integrated Windows Authentication” on page 537
- “Using Access System Password Management” on page 540
- “Using Managed Code and Helper Classes” on page 541
- “Integrating NetPoint with Authorization Manager Services” on page 542
- “Integrating NetPoint with Passport Authentication” on page 542
- “Integrating NetPoint with Smart Card Authentication” on page 542
- “Integrating the NetPoint Security Connector for ASP.NET” on page 543
- “Troubleshooting” on page 543
- “Microsoft Resources” on page 544

Resolving Ambiguous Names

Active Directory running on Windows Server 2003 provides support for ambiguous name resolution (ANR).

ANR is a search algorithm associated with LDAP clients that must be enabled on both the LDAP client and the LDAP server. ANR allows objects to be bound without complex search filters and is useful when locating objects and attributes that may or may not be known by the client.

In NetPoint, ANR is a virtual attribute that does not physically exist in the directory server. NetPoint provides the virtual ANR attribute through the AD_anr.ldif file, which enables NetPoint to interpret ANR requests, map ANR requests to Boolean functions And and Or that expand to a directory-server filter to broaden the search, and send the query to Active Directory.

Note: The AD_anr.ldif file is included in the NetPoint schema installation and must be imported manually. See “Configuring NetPoint for ANR” on page 525.

About ANR Attributes, Searches, and Results

By default, the attributes shown in Table 82 are set for ANR .

Table 82 ANR Attributes

ANR Attributes
displayName
GivenName
LegacyExchangeDN
msExchMailNickname
name
physicalDeliveryOfficeName
proxyAddress
sAMAccountName
Surname

For a search filter such as (anr=von), the server would return objects that matched any of the previously listed attributes equal to von*. When a space is embedded in the search string, the search is divided at the space and an Or search is also performed on the attributes. The server attempts to perform first/last name processing. When there is only one space, the search divides only at the first space.

For example, if the search filter was (anr=Rob AI), the filter expansion would look like the one below.

```
((givenName=Rob AI*)
  (sn=Rob AI*)
  (displayName=Rob AI*)
  (legacyExchangeDN=Rob AI*)
  (name=Rob AI*)
  (physicalDeliveryOfficeName=Rob AI*)
  (proxyAddresses=Rob AI*)
  (saMAccountName=Rob AI*)
  (&(givenName=Rob*)(sn=AI*))
  (&(givenName=AI*)(sn=Rob*))
)
```

Configuring NetPoint for ANR

The attributes used by ANR are configurable. You can specify other attributes to be included in ANR searches by using the Active Directory Schema Snap-in to check the Ambiguous Name Resolution box for the attribute. You can directly set the searchFlags attribute to 5 in the attributeSchema for the attribute you want to include. To include an attribute to be used for ANR, the attribute must also be indexed.

The following task overview outlines the procedures you must complete to enable ANR within NetPoint. After you upload the meta-attribute configuration for ANR into the Oblix configuration branch in the directory server, the ANR attribute should be configured on the profile page and defined as searchable. Attribute access control can also be configured on the same profile page.

Task overview: Setting up NetPoint to use ANR during searches

1. Update NetPoint configuration data to include the ANR meta-attribute details in the Oblix branch of the schema, as described in “Updating NetPoint Configuration Data” on page 526.
2. Make the ANR attribute available to the NetPoint search function in the COREid Server, as described in “Configuring ANR in NetPoint Panels” on page 526.
3. Verify Access Control rights, as described in “Verifying ANR Attribute Access Control” on page 527.
4. Use ANR-to-NetPoint authentication and authorization search filters, as described in “Using ANR in NetPoint Searches” on page 528

Updating NetPoint Configuration Data

You first need to update Oblix data (NetPoint configuration data) to include the ANR meta-attribute configuration information in the Oblix branch. During this procedure, the AD_anr.ldif shown below is executed.

```
#File to load ANR meta-attribute configuration to the Oblix tree.
dn: obattr=anr,obclass=user,OU=Oblix,<domain-dn>
changetype: add
instanceType: 4
distinguishedName:
obattr=anr,obclass=user,OU=Oblix,<domain-dn>
objectClass: oblixmetaattribute
name: anr
obattr: anr
obcardinality: ob_single
obdisplayname: ANR
obdisplaytype: ObDTextS
obsearchable: true
obvisible: true
```

When this procedure is complete, ANR appears as an attribute you can select when configuring NetPoint panels.

To update NetPoint configuration data

1. Locate the AD_anr.ldif file on the machine hosting the COREid Server:
 \COREid_install_dir\identity\oblix\data.ldap\common\AD_anr.ldif.
2. Import the AD_anr.ldif file to the Oblix configuration directory.

For example:

```
D:\data>ldifde -i -f AD_anr.ldif -a
"cn=adminstrator,cn=users,dc=name,dc=company,dc=net" password
```

3. Restart the COREid Server.

Configuring ANR in NetPoint Panels

After you update the NetPoint configuration data with ANR meta-attributes, you are ready to make the ANR attribute available to the NetPoint search function on a Tab (panel) and in the list of searchable attributes in the User Manager Selector.

The procedure below steps you through configuring ANR in NetPoint panels. For more information, see “Configuring User, Group, and Organization Manager” on page 99.

To configure ANR in NetPoint Panels

1. Navigate to the Configure Panel function in the COREid System Console: COREid System Console > User Manager Configuration > Configure Tab > *Desired_Link* > View Object Profile > Configure Panels > *Desired_Link*.

A summary appears listing all attributes for the selected panel.

2. Click the Modify button at the bottom of the summary page to display the Modify Panel page.
3. Click the Add button, then select ANR from the drop-down list in the Attributes column and click Save.



A screenshot of a web form. On the left, there is a dropdown menu with 'ANR' selected. To the right of the dropdown is a text input field containing 'ANR'. Further to the right is a small grey button labeled 'Add'.

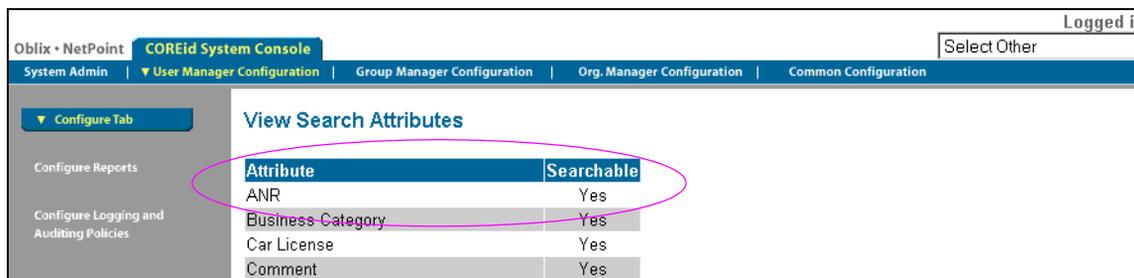
The summary page appears listing all attributes, which should now include ANR.

Next you need to confirm that ANR is a searchable attribute that will appear in the Query Builder's search criteria drop-down list.

4. Click Configure Tab > *Desired_Link*.
5. Click the View Search Attributes button at the bottom of the page.

A list of all search attributes appears.

6. Confirm that ANR is in the list. For example:



The screenshot shows the 'View Search Attributes' page in the COREid System Console. The page has a navigation bar at the top with 'Obliv • NetPoint' and 'COREid System Console'. Below the navigation bar, there are tabs for 'System Admin', 'User Manager Configuration', 'Group Manager Configuration', 'Org. Manager Configuration', and 'Common Configuration'. The 'User Manager Configuration' tab is active. On the left side, there is a sidebar with 'Configure Tab' selected. The main content area shows a table with two columns: 'Attribute' and 'Searchable'. The table contains the following data:

Attribute	Searchable
ANR	Yes
Business Category	Yes
Car License	Yes
Comment	Yes

The 'ANR' row is highlighted with a pink oval.

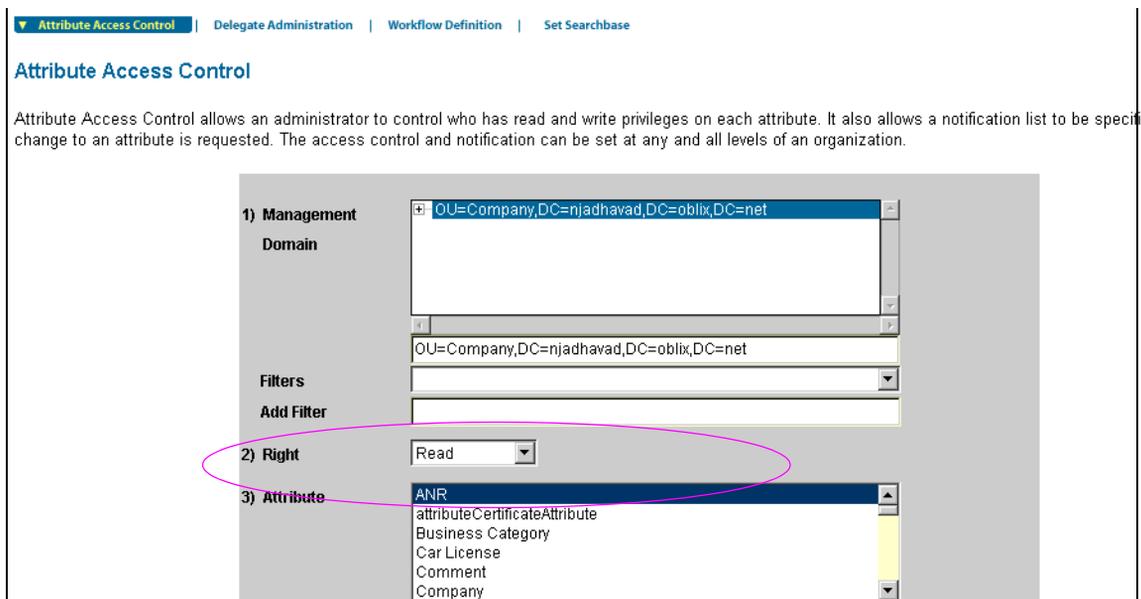
7. Restart the COREid Server.

Verifying ANR Attribute Access Control

By default, the attribute has read rights. The ANR attribute must not have modify rights. The procedure below shows the Access Control rights for the ANR attribute. For more information, see “Setting and Modifying LDAP Attribute Permissions” on page 138.

To verify ANR attribute access control

1. Navigate to the User Manager, as usual.
2. Click the Configuration tab, then click Attribute Access Control.
3. Select ANR from the Attribute list, then verify that it has read rights only.



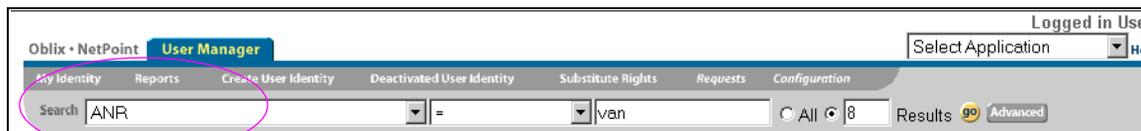
You are ready to use ANR in NetPoint searches.

Using ANR in NetPoint Searches

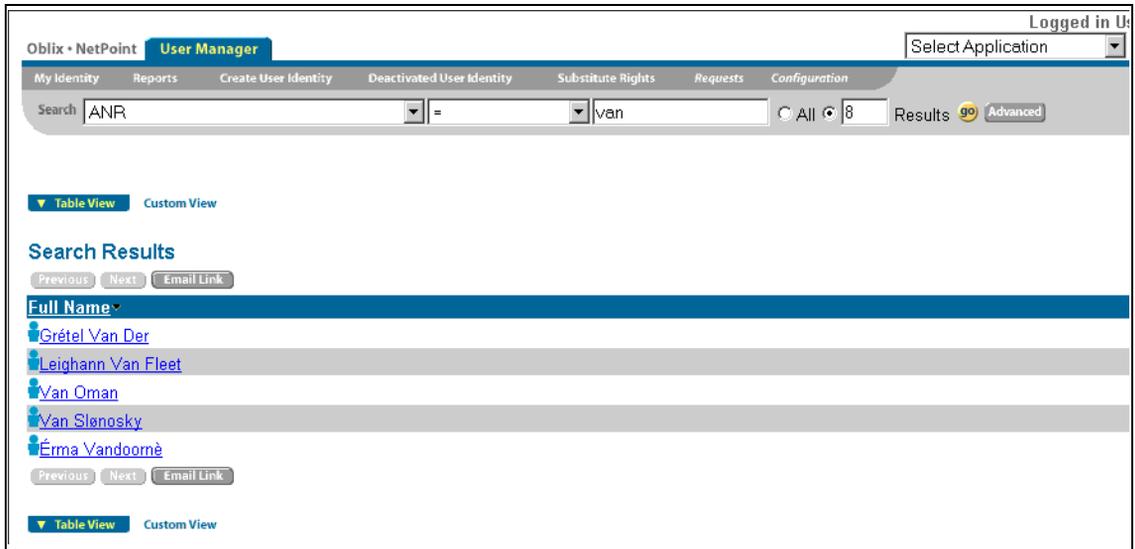
When a user invokes the NetPoint User Manager, they can choose ANR from the search criteria drop-down list to perform a directory search.

To use ANR in a search

1. Navigate to the COREid System main page, as usual.
2. Select the User Manager link to display the Selector.
3. Select ANR from the Search drop-down list, define other search criteria, then enter your condition. For example:



4. Click Go and check your results.



Configuring NetPoint for Dynamically Linked Auxiliary Classes

A structural object class can stand on its own and contains basic attributes required for use within NetPoint applications. Structural object class examples include person and groupOfNames. The person object class may contain attributes such as name, department, employee ID, and email address. A structural object class must be assigned when you create a tab within a NetPoint application.

Auxiliary object classes are *mix-in* classes that can be added to any structural class. You use an auxiliary object class to add a set of related attributes to an entry that already belongs to a structural class. Items such as a billing address, a challenge phrase, a response to a challenge phrase, and so on may be useful for definition in an auxiliary object class.

With Windows Server 2000, Active Directory supported only statically linked auxiliary classes. A statically-linked auxiliary class is one that is included in the auxiliaryClass or systemAuxiliaryClass attribute of an object class's classSchema definition in the schema. It is part of every instance of the class with which it is associated. Using statically-linked auxiliary classes is the default with NetPoint when it is installed with Active Directory. All other directories support only dynamically linked auxiliary object classes.

With a Windows 2003 Server, Active Directory and NetPoint support dynamically linked auxiliary classes. With the schema defined for a particular user, group, or organization, dynamically linked auxiliary classes enable you to store additional attributes with an individual object without the forest-wide impact of extending the schema definition for an entire class. Dynamically linked auxiliary class attributes are mixed in only at runtime.

For example, you can use dynamic linking to attach a sales-specific auxiliary class to the user objects of sales people and other department-specific auxiliary classes to the user objects of employees in other departments. Or you may want to convert a basic group to a mail group by adding specific attributes dynamically.

Task overview: Setting up NetPoint for dynamic auxiliary classes

1. Install and set up NetPoint with dynamic-auxiliary classes enabled, as described in the *NetPoint 7.0 Installation Guide*.
2. Specify additional structural object classes for the Organization Manager, as described in “About Object Classes” on page 60.
3. Configure attributes, as described in “About Object Class Attributes” on page 71.
4. Configure User, Group, and Organization application tabs, as described in “Configuring Tabs” on page 101.
5. Configure User, Group, and Organization profile pages, as described in “Configuring Tab Profile Pages and Panels” on page 113.
6. Define workflows, as described in “Chaining COREid Functions Into Workflows” on page 171.
7. Specify additional auxiliary object classes, as described in “Adding Attributes Dynamically” on page 530.

Adding Attributes Dynamically

The procedure below provides an example only and assumes that you have created a Tab and Panel in the User Manager. Here you will add desired auxiliary attributes dynamically.

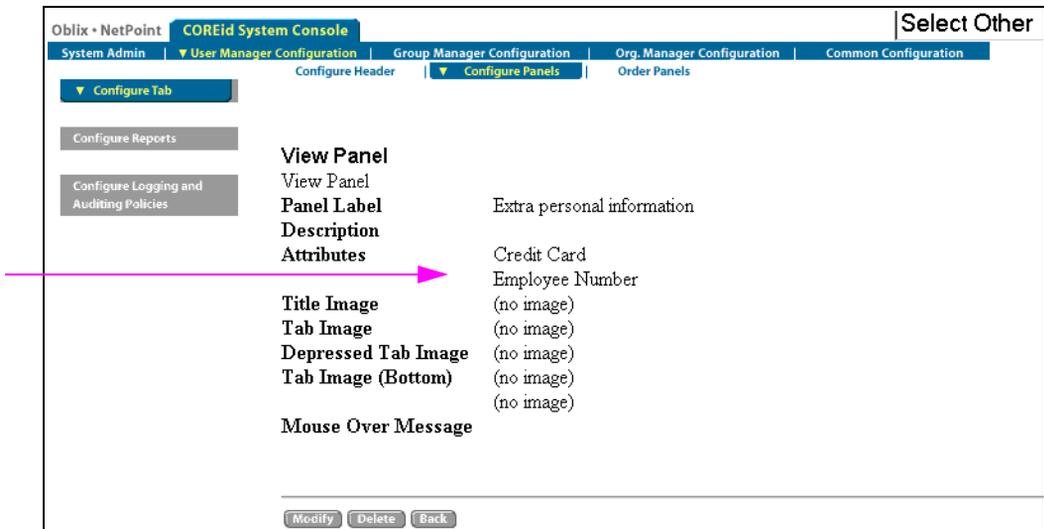
Note: This is only an example. You may be working in the Group Manager or Organization Manager. See also, “Adding Attributes for a Group” on page 531.

To specify additional auxiliary object classes in the User Manager

1. Navigate to the COREid System Console, as usual.
`http://hostname:port/identity/oblix`

1. From the CORE System Console, navigate to User Manager Configuration > Configure Tab > *Link*.
2. Click the View Object Profile button > Configure Panels > *desired_link*.
3. Click the Modify button to display the Modify Panel page.
4. Click the Add button, select one or more attributes from the drop-down list, then click Save.

The View Panel page appears with the attributes you added.



The entry in the directory server has changed, and the new attributes are included.

Adding Attributes for a Group

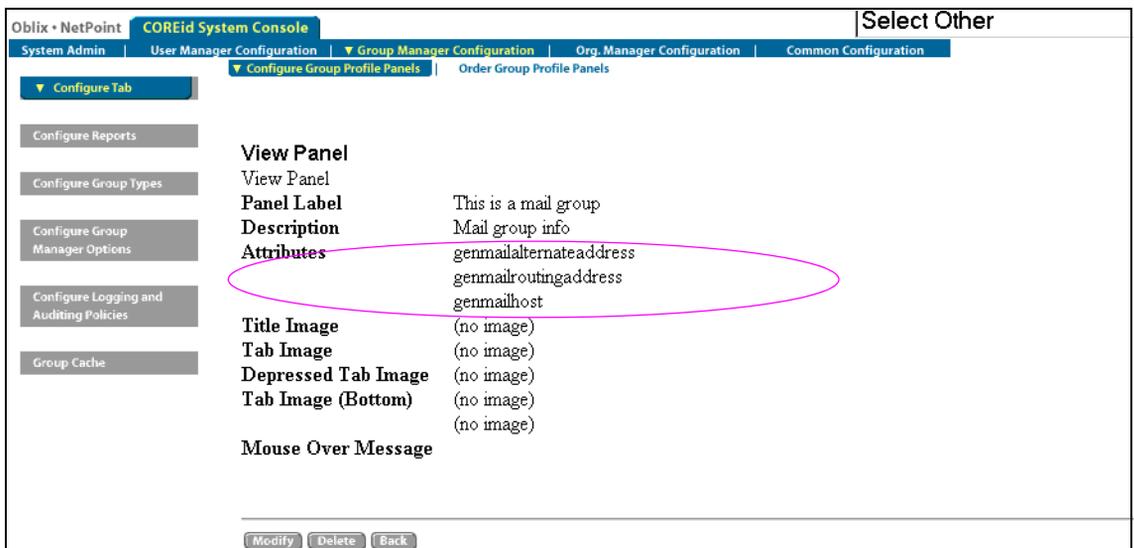
The example in this procedure dynamically converts a single basic group to a mail group by adding attributes, such as:

MailAlternateAddress	Mailhost	MailRoutingAddress
----------------------	----------	--------------------

This example assumes that you have created a Group Panel and a workflow to create a Mail Group. Now you add desired attributes dynamically. This is only an example. You may be working in the User Manager or Organization Manager. See also “Adding Attributes Dynamically” on page 530.

To add attributes to a Group Profile Panel

1. From the CORE System Console, navigate to Group Manager Configuration > Configure Tab > *Group_Link*.
2. Click View Object Profile > Configure Group Profile Panels > *desired_link*.
The View Panel page appears.
3. Click Modify.
The Modify Panel page appears.
4. Click the Add button, select one or more attributes from the drop-down list, then click Save and verify that the attributes you added appear in the View Panel page.



5. Select Group Manager from the Select Application list in the upper right corner.
6. Enter your search criteria in the Selector and click Go.
The results are returned. When you select a Group to review you will notice that the attributes you added dynamically to one group are available only for that group.
7. Click Modify, click the + button, then add a specific value, and save, as usual.

Mail group info

genmailalternateaddress

genmailroutingaddress

genmailhost

Save Group Cancel View Panels

The entry in the directory has also changed. For example, the screen below shows a sample entry before auxiliary classes were added.

LDAP://njadhav.njadhavad.oblix.net/DC=njadhavad,DC=oblix,DC=net

Connection Browse View Options Utilities

ou=Company,dc=njadhavad,dc=oblix,dc=net

- ou=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - OU=Cupertino,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Basic group1k40,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - No children
 - CN=Basic group1k41,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Basic group1k42,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Basic group1k43,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Nested group1k31,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Nested group1k34,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net

Matched DNs:
Getting 1 entries:
>> Dn: CN=Basic group1k40,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net

2> objectClass: top; group;
1> cn: Basic group1k40;
5> ou: Company; Ford; Central; Dealer1k10; Groups;
1> description: Dealer1k10/1 - type=basic, members=12 - discarded=4;
12> member: CN=Jacque Raila,OU=LSales,OU=Los Angeles,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net; <Idp: Binary blob>; <Idp: Binary blob>; CN=Vijay Claveau,OU=Sales,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net; CN=Crissy Kosten,OU=LSales,OU=San

The next screen shows the same entry after auxiliary classes were added.

LDAP://njadhav.njadhavad.oblix.net/DC=njadhavad,DC=oblix,DC=net

Connection Browse View Options Utilities

ou=Company,dc=njadhavad,dc=oblix,dc=net

- ou=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - OU=Cupertino,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Basic group1k40,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Basic group1k41,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Basic group1k42,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Basic group1k43,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Nested group1k31,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net
 - CN=Nested group1k34,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net

Matched DNs:
Getting 1 entries:
>> Dn: CN=Basic group1k40,OU=Groups,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net

1> obgroupstype: genmailgroupauxclass;
1> genmailalternateaddress: foo@xyz.com;
1> genmailroutingaddress: foo@abc.com;
4> objectClass: top; oblixgroup; genMailGroupAuxclass; group;
1> cn: Basic group1k40;
5> ou: Company; Ford; Central; Dealer1k10; Groups;
1> description: Dealer1k10/1 - type=basic, members=12 - discarded=4;
12> member: CN=Jacque Raila,OU=LSales,OU=Los Angeles,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net; <Idp: Binary blob>; <Idp: Binary blob>; CN=Vijay Claveau,OU=Sales,OU=Dealer1k10,OU=Central,OU=Ford,OU=Company,DC=njadhavad,DC=oblix,DC=net; CN=Crissy Kosten,OU=LSales,OU=San

Enabling Fast Bind for NetPoint Authentication

The Active Directory running on Windows Server 2003 provides a concurrent bind (also known as *fast bind*) feature that allows multiple authentications over the same LDAP connection.

NetPoint supports and uses this feature, which provides the following advantages:

- Fast bind permits two threads to request a bind over one connection at the same time.
- Fast bind provides a faster authentication mechanism because it only validates the password and the account flag and does not build a ticket.

The Fast Bind option must be enabled for each database instance, and is located on individual database profiles in the NetPoint System Console.

To configure NetPoint to use a fast bind

1. Navigate to the COREid System Console, as usual. For example:
`http://hostname:port/identity/oblix`
2. Navigate to the Configure Directory Options page: COREid System Console > System Admin > System Configuration > Configure Directory Options

The Configure Directory Server Profiles page is shown below, where you choose the directory profile to modify.

Configure Directory Server profiles
The following contains the Oblix Base and Searchbase settings. Click on the link to change a particular value.

Directory Server

Machine stevef3
Port number 399
Root DN cn=Directory Manager
Root password <Not Displayed>
Search Base o=company,c=us
Oblix base o=Oblix,o=company,c=us
Directory Server Security Mode Open
Disjoint Search Base

The following table contains the list of all Directory Profiles. Click on any link to change a particular profile. You must stop and restart every COREid server before the new values can take effect.

Name	Name Space	Primary Servers	Secondary Servers
<input type="checkbox"/> default-COREid_65b1_6091	o=company,c=us	default	
<input type="checkbox"/> AccessManager_setup_user_profile	o=company,c=us	default	
<input type="checkbox"/> AccessServer_default_user_profile	o=company,c=us	default	

- Click the name of the directory server instance on which you want to enable the Fast Bind feature.

For example:

Access_Manager_setup_user_profile

The Modify Directory Server Profile page appears, and you can locate the instance of the directory server profile to modify near the bottom of the page.

- Locate and click the name of the directory server profile instance you want. For example:

Access Managers

Name	Machine	Port number	Server Type
default	stevef3.oblix.com	399	Primary

Database Instances

Maximum Active Servers

Failover Threshold

Sleep For (Seconds)

Enable Profile

Note: The fields marked with an asterisk(*) are required fields.

5. Locate and check the box beside the Fast Bind option. For example:

The screenshot shows the 'Modify Database Instance' configuration page in the COREid System Console. The page has a left-hand navigation menu with options like 'Configure Admins', 'Configure Styles', 'Import Photos', 'View Server Settings', 'Configure Directory Options' (highlighted), 'Configure Webpass', 'Configure Password Policy', and 'Configure COREid Server'. The main content area contains a form with the following fields and values:

Name*	default
Machine*	stevef3.oblix.com
Port number*	399
Root DN*	cn=Directory Manager
Root password*	
Time Limit	0
Size Limit	0
Flags	<input type="checkbox"/> SSL <input checked="" type="checkbox"/> Refer... <input checked="" type="checkbox"/> Fast Bind (only for AD on Windows Server 2003)
Secure Port number	636
Initial Connections	1

A pink arrow points to the 'Fast Bind' checkbox, which is checked.

6. Click Save.
7. Confirm that the profile has been enabled on the Modify Directory Server profile page.
 Enable Profile
8. Repeat as needed to enable the Fast Bind option for other database instances.

Enabling Impersonation

In a Windows environment, all processes and threads execute in a security context. Impersonation is the ability of a thread to execute in a security context that is different from that of the process that owns the thread. The primary purpose of impersonation is to trigger access checks against a client's identity.

For details about enabling impersonation in NetPoint, which overrides impersonation enabled with IIS, see *NetPoint 7.0 Administration Guide Volume 2*.

Setting Up Integrated Windows Authentication

NetPoint provides support for integrated Windows authentication (IWA). Your environment may include:

- Windows 2000 Server or Windows Server 2003 or Solaris
- Internet Information Services (IIS) 5.x or 6.x
- Active Directory or iPlanet directory server

If the user's directory server has, for example, an NT Logon ID, or if the username is the same everywhere, then a user is able to authenticate into any directory server.

The most common authentication mechanism on Windows 2000 and Windows Server 2003 is Kerberos.

NetPoint's use of IWA is seamless. The user won't notice any difference between a typical authentication and IWA when they log on to their desktop, open an Internet Explorer (IE) browser, request a NetPoint protected web resource, and complete single sign-on.

Process overview: During IWA authentication

1. The user logs in to the desktop machine, and local authentication is completed using the Windows Domain Administrator authentication scheme.
2. The user opens an Internet Explorer (IE) browser and requests a NetPoint protected Web resource.
3. The browser notes the local authentication and sends a token to the IIS Web server.
4. The IIS Web server uses the token to authenticate the user and set up the REMOTE_USER HTTP header variable that specifies the username supplied by the client and authenticated by the server.
5. The WebGate installed on the IIS Web server uses the hidden feature of external authentication to get the REMOTE_USER header variable value and map it to a DN for the ObSSOCookie generation and authorization.
6. The WebGate creates an ObSSOCookie and sends it back to the browser.
7. The NetPoint authorization and other processes proceed as usual.

The maximum session timeout period configured for the WebGate is applicable to the generated ObSSOCookie.

Task overview: Setting Up Integrated Windows Authentication

1. Install a WebGate on the same IIS Web server or servers on which you will set up IWA, as described in the *NetPoint 7.0 Installation Guide*.
 - If you installed the WebGate at the Site level, you should perform the tasks at the Site level.
 - If you have multiple WebGates installed at different virtual sites, you should perform the tasks for each virtual site.
2. Enable IWA on the WebGate, as described in “Enabling IWA on the WebGate Web Server” on page 538.
3. Configure the WebGate to use IWA, as described in “Configuring the WebGate for IWA” on page 539.
4. Create an authentication scheme for IWA in NetPoint, as described in “Creating an IWA Authentication Scheme in NetPoint” on page 539.
5. Test the IWA implementation, as described in “Testing IWA Implementation” on page 540.

Enabling IWA on the WebGate Web Server

The first procedure is to enable IWA on the machine hosting the WebGate.

- If you have installed the WebGate at the Site level, you should perform the tasks at the Site level.
- If you have multiple WebGates installed at different virtual sites, you should perform the tasks for each virtual site.

To enable IWA on the machine hosting the WebGate

1. Start the Internet Services Manager on the machine hosting the WebGate: Start > Programs > Administrative Tools > Internet Services Manager
2. Right-click the Default Website (or the name of Web server if you changed the name of the Default Website), then select Properties.

Note: If you installed WebGate at the Site level, right-click the Site then select Properties.

3. Click the Edit button beside Master Properties.
4. Click the Directory Security tab, then click Edit beside “Anonymous access and authentication control.”
5. Disable Anonymous Access on the IIS Web Server.
6. Enable Integrated Windows Authentication.

7. Click OK, then click OK again.
8. Restart the IIS Web server.

Configuring the WebGate for IWA

Next, you perform the following procedure for the NetPoint access virtual directory that you have created on this machine.

To configure the WebGate for IWA

1. On the machine you have set up for IWA, navigate to the file below:
`\WebGate_install_dir\access\oblix\apps\webgate\WebGateStatic.lst`
2. Change the `UseIISBuiltinAuthentication` parameter value to `true`, and save your changes.

For example:

```
UseIISBuiltinAuthentication:true
```

3. Restart the IIS Web Server.

Creating an IWA Authentication Scheme in NetPoint

You must create an IWA authentication scheme for NetPoint to use a specific challenge method, challenge parameter, and plug-in, as described below.

To create an IWA authentication scheme in NetPoint

1. Navigate to the Access System Console, as usual. For example:
`http://hostname:port/access/oblix`
2. Navigate to the Authentication Management page and click Add: Access System Console > Access System Configuration > Authentication Management > Add.
3. Create an Integrated Windows Authentication scheme.

For example:

Name : *Integrated Windows Authentication*

Description : *This scheme is Integrated Windows Authentication, using the built-in Windows authentication mechanism.*

Level : 1

Challenge Method : Ext

Challenge Parameter : creds:REMOTE_USER

SSL Required: No

Challenge Redirect

4. Click the Plug-Ins tab, then click Modify.

5. Select the plug-in name from the drop-down list, enter your plug-in parameters and click Add, then save when you are finished.

For example:

Plugin(s)

Plugin Name	Plugin Parameters
credential_mapping	obMappingBase=<"Domain name">,obMappingFilter="(&(objectclass=user)(samaccountname=%REMOTE_USER%))"

6. Save the authentication scheme and protect resources using this scheme, as usual.

Testing IWA Implementation

It is always a good idea to test the implementation before you roll it out.

To test IWA

1. Log in to the machine as someone who is a user of both NetPoint and the Windows operating system.
2. Enter the URL of the protected resource.

Using Access System Password Management

When using the Access System Password Management feature with an Active Directory forest, note the following:

- The Change on Reset, Password Expiration, and Password ExpirationWarning features will work.
- The Number of Retrieves feature will not work.

This limitation applies only if you are using the LDAP mode for Password Management in the Access System and only if you are using Active Directory in a forest configuration.

Using Managed Code and Helper Classes

The .NET Framework, supported by NetPoint v6.5 and later, provides an object-oriented programming environment to guarantee the safe execution of code and to eliminate performance problems in scripted environments. In the .NET Framework, code that targets the runtime is called managed code.

In addition, MANAGEDLIB actions offer the benefits of managed code, including:

- **Language Choice**—You can write your plug-ins in VisualBasic, C#, Managed C++ (MC++), Java, or PERL.
- **Language Integration**—You can combine MIL modules compiled from different source languages into one assembly or plug-in.

This provides the plug-in writer with a wider range of language choices for plug-in development.

- **Support for Memory Management**—The common language runtime (CLR) provides garbage collection, freeing the plug-in writer from most memory management.

The garbage collector returns memory to the heap when that memory is no longer referenced. However, the plug-in writer should ensure that there are no dangling references to objects. If there are dangling references, garbage collection will not occur for the unused memory.

- **.NET Framework Support**—The .NET framework SDK contains a wide range of functionality. This may reduce the need for third-party support in plug-in code.

NetPoint can use and call APIs in many languages, and NetPoint APIs can now be written in managed code using a variety of languages, including C, Managed C++ (MC++), and Visual Basic.Net.

For more information about managed code, managed helper classes, and NetPoint, see the *NetPoint 7.0 Developer Guide*.

Integrating NetPoint with Authorization Manager Services

NetPoint provides an authorization plug-in that uses the Microsoft Windows Server 2003 Authorization Manager (AzMan) services to make authorization decisions for Access Server clients, including WebGates and callers of the Access Server API.

See the *NetPoint Integration Guide* for details about configuring a NetPoint policy domain for the NetPoint AzMan plug-in.

Integrating NetPoint with Passport Authentication

Passport authentication, now available for IIS 6.0, enables Active Directory user objects to be mapped to their corresponding Passport identification, if it exists. A token is created by the local security authority (LSA) for the user and is set by IIS 6.0 for the HTTP request. Internet users who have a corresponding Passport identification may now use their passport to access resources as if they were using their Active Directory credentials.

For information about integrating NetPoint with Passport authentication, see the *NetPoint Integration Guide*.

Integrating NetPoint with Smart Card Authentication

NetPoint supports smart card authentication with Active Directory and IIS Web servers in homogeneous Windows environments. Using a smart card provides a stronger form of authentication than a username and password alone because it is based on *something the user knows* and *something the user has*.

- *Something the user knows* is the user's secret personal identification number (PIN), similar in concept to a personal bank code PIN.
- *Something the user has* is the cryptographically-based identification and proof-of-possession generated by the smart card device that you insert into the smart card reader attached to a computer.

See the *NetPoint Integration Guide* for details about configuring integrating NetPoint with smart card authentication.

Integrating the NetPoint Security Connector for ASP.NET

NetPoint supports the ASP.NET component of the Microsoft .NET Framework, which developers can use to build, deploy, and run Web applications and distributed applications. The NetPoint Security Connector for ASP.NET supports and enhances native .NET role-based security.

See the *NetPoint Integration Guide* for details about how to use the NetPoint Security Connector for ASP.NET to instantiate a new `OblixPrincipal` object and populate it with roles (NetPoint authorization rules) and the native `WindowsPrincipal` object.

Troubleshooting

Active Directory Search Halts

Symptom—400 policy domains were created in NetPoint, each with 10 resources and 10 policies. The `limitAMPolicyDomainResourceDisplay` is set to true in the Access Manager `globalparams.lst` file. When the Search icon is selected, an error page appears stating “The following messages were produced by the product. Please contact your webmaster to fix the problem.”

Cause—The number of policy domains exceeds the current limit.

Solution—Do not exceed 350 policy domains with Active Directory.

Microsoft Resources

Active Directory Home Page

<http://www.microsoft.com/windows2000/technologies/directory/ad/default.asp>

ADSI Overview

<http://www.microsoft.com/windows2000/techinfo/howitworks/activedirectory/adsilinks.asp>

Active Directory Programmers Page

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netdir/adsi/active_directory_service_interfaces_adsi.asp?frame=true

ADSI Programmers Page

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netdir/adsi/active_directory_service_interfaces_adsi.asp?frame=true

Index

Symbols

.NET Framework Support 541

A

About link 39

access control

reserved words 165

sample gscacparams.xml file 163

simplified access control, group attributes 163

Access Manager

Active Directory forest, configuration 496

ADSI configuration file location 508

globalparams.lst file 508

access rights

administrator rights, changing 137

Access Server

Active Directory forest, setting up 519

password, changing 366

Access Server, ADSI

ADSI authentication 507

adsi_params.lst file 509

configuration file location 508

Access Server, installation and configuration

SDK, configuring 333

Access Server, LDAP authentication

ADSI, enabling for 514, 521

Global Catalog requirement 507

Access Server, transport security

changing modes, considerations 350

signed certificate, installing 358

updating definitions 349

Access System

Active Directory single sign-on requirements,
accommodating 495

authentication compatibility 507

binding mechanisms 507 to 508

directory profiles and user access 496

logging in 36, 41

Number of Retrieves feature 540

Access system

cert mode, migrating to 356

open mode, migrating to 353

Simple mode, migrating to 354

Access System, Active Directory forest

Access Server setting up 519

Access System modifications 517

configuring for Access Manager 496

Access System, ADSI

configuration process overview 513

support for 506

Access System, password management
authentication schemes, modifying 329 to 330
password policies, role of 320
redirect URL, modifying 331

Access System, single sign-on

Active Directory requirements 495

redirect URL, entering 331

Access Testing, definition 368

AccessGate

command line, modifying from 315

AccessGate, transport security

signed certificate, installing 358

updating definitions 349

Accounts

creating accounts in external applications 253

See also, Provisioning

Activate workflow action

about 186

Active Directory

ActiveDirectory parameter 509

ADSI, advantage of using 499

attribute availability 71

COREid Server, binding mechanism 502

credential_mapping plug-in 494

directory profile, specifying 295

directory server profiles 488

indexed searches 497

multiple instances and dynamic filter attribute 146

Oblix data replication, recommendation 500

single sign-on requirements, accommodating 495

technical resources, location of 498, 544

timeouts 520 to 521

which contains, search filter 497

Active Directory forest

Access Server, setting up 519

Access System modifications 517

diagram 488

disjoint searchbase, adding 131, 489

pageSize parameter, changing 515

password features, consideration 540

SSL requirement 501

user domain, determining 496

Active Directory Search Halts 543

Active Directory Services Interface. *See* ADSI

AD_anr.ldif file 524

Adding Attributes Dynamically 530

Adding Attributes for a Group Dynamically 531

administrator rights, changing 137

ADSI

See also Active Directory; Active Directory forest

additional directory profiles, enabling for 511 to 512

agent/domain relationship 502

global_params.xml file 503

LDAP authentication 502

"pure" ADSI setup 501

SSL requirement 501

ADSI and Access Manager

configuration file location 508

- ADSI and Access Server
 - ads_i_params.lst file 509
 - configuration file location 508
 - LDAP authentication, enabling 514, 521
- ADSI and Access System
 - configuration process overview 513
- ADSI, authentication
 - about 501
 - Access System authentication 507
 - LDAP authentication 502
 - LDAP authentication, Access Server 514, 521
- ads_i_params.xml file
 - Access Server configuration 509
 - COREid System 504 to 505
 - page size, changing 515
- ads_iCredential parameter
 - in Access System configuration 508
 - Access System, values of 509
 - in COREid System configuration 504
 - COREid System, values of 505
- ads_iEnabled parameter 503, 509
- ads_iPassword parameter 505, 509
- ads_iUPN parameter
 - Access System and explicit bind 508
 - Access System setting for implicit binding 510
 - Access System, values of 509
 - COREid System, implicit bind 505
- Allow users to override defaults through URL
 - parameters, display option 111
- Ambiguous name resolution 523
- ANR attribute 524
- application accounts
 - provisioning 254
- Application navigation bar 38
- Approval workflow action 186
- ASP model, delegated administration 51
- ASP.NET 543
- asynchparams.xml 228
- Asynchronous mail option 282
- asynchronous operations, performing in workflows 228
- Attribute data type 73
- attribute semantic types
 - determining 82
 - Group Manager 82
 - Organization Manager 82
 - User Manager 82
- attributes
 - auxiliary object types 109
 - changing 179
 - common attributes, modifying 71 to 83
 - data types 82
 - multiple values 82
 - NDS to LDAP mapping considerations 82
- attributes, access control
 - configuring group access 138
 - evaluation, User Manager 142
 - reserved words 165
 - sample gscacparams.xml file 163
- attributes, Active Directory
 - availability for Active Directory users 71
- attributes, configuring
 - derived attributes, configuring 95, 96
- attributes, derived
 - about configuring 94
 - assigning rights 97
 - configuring 95, 96
 - Group Manager tab, assigning to 96
 - User Manager tab, assigning to 96
- attributes, Group Manager
 - access rights, specifying 138
 - additional attributes 80
 - additional Display Type fields 83
 - modifying 80
 - multiple values, use of 82
 - simplified access control, about 163
- attributes, Organization Manager
 - conflicting tab configurations, resolving 71
- attributes, reporting
 - disallowed data types, Group Manager 82
 - disallowed data types, User Manager 82
 - User Manager tab attribute reports 156 to 159
- attributes, rights
 - derived attributes, assigning rights 97
 - User Manager access rights, specifying 138 to 141
- attributes, search results
 - search functionality in Organization Manager, enabling 106
- attributes, User Manager
 - access control, evaluation 142
 - access rights, specifying 138 to 141
 - additional attributes, modifying 82
 - associating with rights 142
 - configuration keyboard shortcuts 141
 - conflicting tab configurations, resolving 81
 - multiple values, use of 82
 - rights, assignment methods 140
 - trustee (participant) access 139
- attributes, workflows
 - attribute properties, configuring 205 to 206
 - Change Attribute workflow 210
 - saving, restriction on 205
 - workflow steps, selecting for 204
- auditing
 - audit database 409
 - audit database, creating in MySQL 423
 - audit database, creating in SQL Server 423
 - audit schema 424
 - audit schema, uploading to MySQL 426
 - audit schema, uploading to SQL Server 425
 - audit-to-database architecture 405
 - audit-to-database requirements 403
 - audit-to-database set up 416
 - configuring for each Access server 447
 - configuring for each COREid server 439
 - configuring the Master Audit Policy 440
 - connecting NetPoint servers to the audit database 428

- controlling output 399
- Crystal repository 410
- database auditing 416
- dynamic 398
- editing ODBC.ini 430
- file-based auditing 412
- MyODBC driver for MySQL 429
- MySQL set up 419
- NetPoint Audit Reports 411, 451
- Oblix repository 410
- ODBC data source definition, creating in UNIX 430
- ODBC data source definition, creating in Windows 431
- ODBC data source definitions 406
- ODBC drivers 407
- orMap.ini, editing 454
- performance considerations 396
- RDBMS instance, primary 435
- RDBMS instance, secondary 436
- RDBMS profiles 408, 433
- remote connections, enabling for MySQL 422
- replacing the audit output format string (Access System) 448
- replacing the audit output format string (COREid System) 440
- required special components 404
- security considerations 396
- setting audit options 400
- SQL Server set up 418
- static 397
- transactional support for MySQL 421
- User access privilege reports 449
- auditing policies
 - User Manager 153 to 154
- Auditing, definition 368
- authentication
 - COREid System compatibility 507
 - directory profile, role of 496 to 497
- authentication schemes
 - password policy, including 329 to 330
 - validate_password plug-in 329
- authorization
 - directory profile, role of 496 to 497
- Authorization Manager Services 542
- auxiliary class
 - associating with a structural class 70
- Auxiliary Classes
 - Dynamically-Linked 529
- auxiliary object class
 - about 63, 529
 - associating with organization 108
- asynchronousSearch parameter 505, 510

B

- back-end applications
 - provisioning accounts 253

- Basic Over LDAP scheme, example 329
- binding mechanisms
 - Access System 507
 - changing after initial setup 504
 - COREid Server to Active Directory 502
- binding, explicit
 - Access System 507 to 508
 - adsiUPN parameter, setting of 508
 - COREid System 503
- binding, implicit
 - Access System 507
 - adsiUPN parameter, Access System 510
 - adsiUPN parameter, COREid System 505
 - COREid System 502
 - useImplicitBind flag 502, 507
 - useImplicitBind parameter, Access System 509
 - useImplicitBind parameter, COREid System 505
- bug reports
 - configuring email address for 280
- BypassAccessControlForDirAdmin parameter 137

C

- Cert mode
 - See also* transport security certificate; transport security integration; transport security modes
 - installing a signed certificate 347
 - process overview 344
 - server definitions, updating 343
 - transport security mode password, changing 363
 - WebPass definitions, updating 343
- certificate management
 - certificate request, generating 356
 - internal CA, generating through NetPoint 354
 - signed certificate, installing 358
- Certificate Processing Server (CPS)
 - certificate request, generating 346
- challenge and response
 - See also* authentication; lost password management; passwords
 - attributes, about encryption 327
 - Response attribute encryption 327
- Change Attribute workflow
 - See also* subflows; workflows
 - creating 210
- Change Information and Approval workflow action 186
- Change Information workflow action 186
- changing an attribute 179
- Classic Style, customizing 271
- cloning 366
- Commit workflow action 186
- Configure Group Types 109
- Configure Session Timeout option 278
- ConfigureAccessGate tool 315
- Configuring ANR in NetPoint Panels 526
- Configuring NetPoint for ANR 525
- Configuring NetPoint for Dynamically-Linked Auxiliary

- Classes 529
- Configuring the WebGate for IWA 539
- contact information 19
- container limits
 - about 165
 - adding 166 to 167
 - copying 168
 - viewing 165
- COREid Server
 - Active Directory, binding mechanisms 502
 - caches, managing 283
 - generating internal certificate 354
 - restoring 285 to 287
 - server settings, viewing 277
- COREid Server and WebPass
 - associating WebPass instance 319
 - compatibility with WebPass transport security modes 313
 - instance, disassociating 319
- COREid Server parameters
 - viewing 288
- COREid Server, transport security modes
 - See also* transport security certificate; transport security integration; transport security modes
 - about 337
 - certificate password, changing 363
 - certificate request, generating 346
 - server definitions, updating 343
 - signed certificate, installing 347
 - Simple mode, generating certificate 345
- COREid System
 - authentication compatibility 507
 - bug reports, configuring email address for 280
 - common object classes 65 to 70
 - email address configuration 280
 - global_params.xml file 503
 - licenses, configuring 279 to 280
 - logging in 35
 - mail servers, configuring 282 to 283
 - password policies, role of 320
 - Response attribute encryption 327
 - session timeout, configuring 278 to 279
 - user feedback email, configuring 280
 - Web master, configuring email address for 280
 - workflows, about creating 171
- COREid system
 - transport security 339
 - transport security mode
 - cert,changing to 346
 - simple,changing to 345
- COREid System and Active Directory
 - single sign-on requirements, accommodating 495
- COREid System and ADSI
 - adsi_params.xml file 504 to 505
 - authentication, types of 502
 - default directory profile, enabling ADSI 510
- COREid System applications
 - See also* Group Manager; Organization Manager;

- User Manager
 - appearance, about 266
 - application-object class relation 66
- COREid System styles
 - See also* styles, customizing
 - customized styles, about 266
 - new styles, creating 269 to ??
- Create Object workflow, first step actions 202
- Create User workflow
 - See also* subflows; workflows
 - subflow sample 189
 - workflow sample 171
- Creating an IWA Authentication Scheme in NetPoint 539
- credential_mapping plug-in
 - Active Directory configuration 494
 - examples 331
- Crystal Reports 451
 - required patch 452
- custom integrations 254
- Customize Email function 280
- Customize Styles function. *See* styles, customizing

D

- data type 73
- data types
 - disallowed data type attribute reports, Group Manager 82
 - disallowed data type attribute reports, User Manager 82
 - Group Manager attributes 82
 - Organization Manager attributes 82
 - User Manager attributes 82
- deactivated users
 - Deactivate workflow action 186
- default_subscription_policy parameter 198
- delegated administration
 - ASP model 51
 - delegated administrators, adding 52 to 54
 - extranet model 49
 - intranet model 50
 - methods for assigning rights, listed 54
 - rights, listed 53
- delegated models, Intranet 50
- Delete workflow action 187
- Deleting Object Classes 71
- derived attributes
 - assigning rights 97
 - configuring 94 to 95
 - Group Manager tab, assigning to 96
 - User Manager tab, assigning to 96
- Diagnostics, definition 368
- directory information tree (DIT)
 - controlling access to 127
 - directory profiles, role of 291
- directory profiles
 - ADSI support, Access System 506

- ADSI, enabling for additional profiles 511 to 512
- authentication vs. authorization 496 to 497
- creating 293 to 298, ?? to 298
- default profile, about 291
- naming recommendation 511
- role in determining user domain 496
- supported operations 292, 295, 512
- directory server instance
 - adding 301
 - definition 301
 - deleting 305
- directory server profiles
 - about 290
- directory servers
 - directory server profile, role of 290
 - server specification in directory profile setup 294
- Disable workflow action 187
- disjoint searchbase 489
- Disjoint_domain 489
- display types 78
 - User Manager 83
- DIT. *See* directory information tree (DIT)
- domains
 - ADSI agent/domain relationship 502
- dynamic audit reports 369
- dynamic groups
 - expanding 161
- Dynamically-Linked Auxiliary Classes 529

E

- Email 188
- email
 - destinations, customizing 280 to 281
 - user feedback configuration 280
- Enable workflow action 187
- Enabling Fast Bind for NetPoint Authentication 534
- Enabling IWA on the WebGate Web Server 538
- Error Report workflow action 187
- exit_condition workflow parameter 246
- explicit binding
 - Access System 507 to 508
 - adsiUPN parameter, setting of 508
 - COREid System 503
- External Action
 - compared to Identity Event Plug-in API action 245
 - workflow action 187
- External applications
 - creating accounts in external applications 253
 - See also*, Provisioning
- extranet model, delegated administration 49

F

- failover
 - Active Directory configuration requirements

- 520 to 521
 - primary server, role of 297
- Fast Bind 534
- Filtered Queries, definition 368
- forceCommit workflow parameter 246

G

- Global Catalog
 - Access Server, LDAP authentication 507
- global parameters
 - Access Manager configuration 508
 - COREid System, global_params.xml file, about 503
 - single sign-on requirements, Active Directory 495
- globalparams.xml 188
- Group
 - creation using NDS 147
- group expansion operation 161
- Group Manager
 - See also* User Manager; Organization Manager
 - auxiliary object types, managing 109
 - data types 82
 - display options 111
 - display types 83
 - features displayed, controlling ?? to 111
 - group creation rights 146
 - group membership 161
 - Group Types, role of 109
 - Semantic Types, listed 82
- Group Manager attributes
 - additional attributes, configuring 80
 - additional display type fields 83
 - derived attributes, assigning to a tab 96
 - modifying 80
 - multiple values, use of 82
- Group Manager workflows
 - actions, table of 186 to 188
 - first step, valid action 202
 - Group Type designation 195
 - Group Type panel, role in running workflow 209
 - Oblix Advanced Group 197
 - workflow types 179
- group members
 - about managing 147
 - adding or deleting 150
- Group Profile panels
 - about 115
- Group Type panels
 - about 109
 - adding 122 to 123
 - parameters, modifying 122
 - viewing 121
- Group Types
 - about 109
 - associating auxiliary object classes 123
 - deleting 122
 - multiple group type panels, example 109

- role of 109
- Groups
 - right to create 146
- groups
 - adding 179
 - dynamic/static group consideration 161
 - multiple Active Directory instances, creating 146
 - multiple groups, subscribing to 153
 - NDS, displaying members 147
 - privileges on attributes, modifying 138
 - static membership list, generating 161
- gscacparams.xml, sample file 163

H

- Header panel, configuring users 113, 114
- Helper Classes 541
- HTML compared to MHTML 281

I

- Identity Event Plug-in API
 - compared to External Action 245
 - workflow custom actions 244
- Identity Server
 - viewing WebPasses associated with 318
- images
 - tab image, User Manager 103
- implicit binding
 - Access System 507
 - adsiUPN parameter, Access System 510
 - adsiUPN parameter, COREid System setting 505
 - COREid System 502
 - useImplicitBind flag 502, 507
 - useImplicitBind parameter, Access System 509
 - useImplicitBind parameter, COREid System 505
- incoming requests 230
- initialStep workflow parameter 246
- Initiate workflow action 187
- installation
 - cloning 366
 - silent mode
 - cloning and synchronizing 366
 - synchronizing 366
- Installing MySQL (UNIX 419
- intranet model, delegated administration 50
- IWA 537

L

- Language evaluation order 275
- LDAP
 - Active Directory timeout configuration requirements for Access Server 520 to 521
 - authentication, enabling for Access Server 514, 521

- Global Catalog requirement, Access Server 507
- Novell Directory Server attribute and object class
 - name mapping 82
 - object classes 65 to 70
 - organization objects 101
- LDAP directory 254
- LDAP directory, deleted users and administration rights 47
- LDAP filters
 - about 83
 - advanced
 - using Query Builder 135
 - dynamic/static group expansion consideration 161
 - rules, defining 83
 - static membership list (group), generating 161
 - substitution syntax 89
 - using Query Builder 132, 133
- license keys
 - COREid System, enabling 279 to 280
 - COREid System, updating 280
- licenses, configuring 279
- lists, compared to rules 83
- localized access. *See* searchbase, setting
- localizing, tabs 104
- locations, creating workflows 250
- logging
 - configuring logs in the COREid system console 390
 - default log settings 388
 - determining whether logging is active 384
 - embedded comments in log configuration files 377
 - log configuration files 375
 - log configuration parameters 386
 - log handlers 385
 - log levels 373
 - log threshold level 384
 - log writers 380
- Logging out 41
- logging policies
 - User Manager 153 to 154
- Logging, definition 367
- lost password management
 - challenge and response attributes, about encryption 327
 - enabling or disabling 328
 - implementing as a portal insert 327
 - implementing for COREid System ?? to 328, ?? to 328
 - redirect URL for Access System 330

M

- mail servers, configuring for COREid System 282 to 283
- Main body 38
- Manage Group Members page 147
- Managed Code 541
- Master Identity Administrator
 - role of 45
- Master Identity Administrators

- configuring 46
- MHTML compared to HTML 281
- Microsoft Resources 544
- MIIS 254
- MIME encapsulation, aggregate documents 281
- Modify Attributes feature 71
- Modify Search Attributes function 81
- Monitoring, definition 367
- mouse over message
 - User application tab 103

N

- name spaces, problem with overlapping 294
- NetPoint
 - description of screens 38
 - documentation 18
 - internal certificate (CA), generating 354
 - screens
 - common elements 38
- NetPoint administrators
 - configuring 46
 - deleting administration rights 47
 - description of 44
 - role of 43
- NetPoint applications
 - configuration task overview 25
- NetPoint Audit Reports 411, 451
- NetPoint Audit Reports, definition 368
- NetPoint Security Connector for ASP.NET 543
- Novell Directory Server (NDS)
 - clearing members from screen, preventing 147
 - NDS to LDAP mapping considerations 82
- Number of Retrieves feature 540

O

- ObAttributeDefinition statement 260
- ObClassDefinition statement 261
- obgroupdynamicfilter attribute 161
- obgroupexpandeddynamic attribute 161
- obgroupSubscriptiontype attribute 197
- object classes
 - changing the structural object class 68
 - deleting 71
- object classes, auxiliary
 - associating with organization 108
 - group type, role of 109
 - Group types, associating with 123
 - Group types, role of 109
- object classes, configured
 - adding ?? to 70
 - application-object class relation 66
 - modifying 67
 - NDS to LDAP mapping considerations 82
 - object class attributes, types, and kinds 66

- Object template
 - file format 257
- Object templates
 - .tpl file location 257
 - configuration 257
 - how they are used in workflows 256
 - introduction 254
 - See also, Provisioning
- object templates 254
 - configuring 257
 - file format 257
 - sample file 260
 - schema file 256
- objects
 - creating 180
- Oblix data
 - location of 503
- oblixadvanced Group object class 197
- oblixadvancedgroup 65
- oblixadvancedgroup object class 65
- occurrence workflow parameter 246
- Online Help 38
- Open mode
 - See also transport security certificate; transport security integration; transport security modes
 - changing to Simple or Cert mode 356
 - migrating to 353
 - process overview 344
- options file
 - cloning and synchronizing 366
- Oracle contact information 19
- Organization Manager
 - See also Group Manager; User Manager
 - about 101
 - data types 82
 - display types 83
 - search functionality, enabling 106
 - Semantic Types, listed 82
- Organization Manager attributes
 - conflicting attribute configurations, resolving 71
 - search functionality, enabling 106
- Organization Manager container limits
 - about 165
 - adding 166 to 167
 - viewing 165
- Organization Manager tabs
 - adding, deleting, or modifying 105 to ??
 - resequencing 112
- Organization Manager workflows
 - valid actions for first step 202
 - workflow actions, table of 186 to 188
 - workflow types 180
- organization objects 101
- Organization Profile panels
 - configuring 116
- outgoing requests 230

P

- pageSize parameter 505, 509, 515
- panels, User Profile
 - adding or modifying 116 to 118
 - resequencing or deleting 120
- Passport Authentication 542
- Password Change Redirect URL 330
- Password Expiry Warning Redirect URL 332
- Password Management 540
- password policies
 - about 320
 - creating 322 to 326
 - Defaults button 322, 326
 - deleting 327
 - directory server specification 294
 - expiration, notification of 324
 - history 325
 - length requirement 323
 - lockout duration 325
 - lost password management, implementing for COREid System 328
 - nonalphanumeric character password requirement 324
 - number of login tries 325
 - overview 320
 - parameters, modifying 326
 - policy evaluation order 321
 - redirect URL, setting up 332
 - uppercase and lowercase character requirements 323
 - validate_password plug-in 329
 - viewing 322
- password policies, Access System
 - authentication scheme, modifying 329 to 330
 - redirect URL, entering 331
- passwords
 - Number of Retrieves feature 540
- Photos 143
 - referencing in a file system 145
 - storing in a directory 143
- photos
 - storage locations for 143
- post-notification 188
- pre_action workflow parameter 246
- primary server and failover 297
- privileges
 - administrator privileges, changing 137
 - delegated administrator rights, listed 53
 - methods for assigning rights, listed 54
 - setting or modifying by group 138
 - setting or modifying by user 138, 141
 - simplified access control, group attributes 163
 - specifying group privileges for attributes 138
 - User Manager access rights, specifying 138 to 141
 - User Manager, associating with 142
- privileges, User Manager
 - about delegating 47
 - ASP delegated administration example 51
 - delegated administrators, adding 52 to 54

- extranet delegated administration example 49
- intranet delegated administration example 50
- Procedure
 - To add a COREid Server 285
 - To add a derived attribute to an application tab 96
 - To add a disjoint searchbase for a disjoint domain 131
 - To add a style 269
 - To add a tab 105
 - To add a WebPass 312
 - To add an object class 69
 - To add attributes to a Group Profile Panel 532
 - To add disjoint searchbase for disjoint_domain 489
 - To add group members 150
 - To add or delete a log file configuration 392
 - To add roles to a workflow definition applet 247
 - To add, modify, or delete a Group Type panel 122
 - To allow a user to perform an asynchronous operation 228
 - To archive a workflow 234
 - To assign or remove a substitute 55
 - To associate a COREid Server and WebPass 319
 - To associate a subflow with a workflow 211
 - To associate an ADSI agent with every domain 502
 - To associate an auxiliary object class with a tab for the User and Organization Manager 108
 - To assume an identity 56
 - To begin a new workflow 196
 - To build a complex filter 135
 - To change a style 271
 - To change a style name 271
 - To change certificate password for Access System 364
 - To change certificate password for COREid 363
 - To change COREid Server transport security 343
 - To change the order in which panels are displayed 120
 - To change to Cert security mode 356
 - To change to Open security mode 353
 - To change to Simple security mode 354
 - To change transport security between the Access Manager and directory server 361
 - To change transport security mode password 317
 - To change transport security password 366
 - To change WebPass transport security 343
 - To configure a derived attribute 95
 - To configure a GIF image display type 93
 - To configure a mail server 282
 - To configure a report 156
 - To configure a role 247
 - To configure an attribute 82
 - To configure ANR in NetPoint Panels 527
 - To configure attribute properties 205
 - To configure collection of SNMP statistics 471
 - To configure Group search read operations 490
 - To configure language-specific workflow panel information 243
 - To Configure NetPoint to use a fast bind 534
 - To configure photos for importing to a directory 144
 - To configure SSO with Active Directory 495
 - To configure the Access Server SDK 333

- To configure the credential_mapping plug-in 494
- To configure the header panel 114
- To configure the length of a user's COREid system session 278
- To configure transport security 316
- To configure WebGate for IWA 539
- To copy a workflow 237
- To copy container limits from one domain to another 168
- To create a derived attribute 95
- To create a directory server profile 293
- To create a filter 88
- To create a password policy 322
- To create a self-registration workflow 248
- To create a static filter 90
- To create a static search filter using a wild card 90
- To create a subflow 210
- To create an IWA authentication scheme 539
- To create or add a panel 116
- To create this workflow 209
- To create, view or modify a panel 118
- To create, view, and modify localized tab configuration 104
- To create, view, or modify localized attribute display names 86
- To customize email destinations 280
- To customize search results 149
- To define a list 85
- To define a rule 84
- To define a self-registration workflow using the Quickstart tool 193
- To define a workflow target 200
- To define a workflow using the QuickStart tool 191
- To define subsequent steps in a workflow 207
- To define the first step in a workflow 202
- To delegate administration 52
- To delete a container limit 169
- To delete a COREid Server's parameters 288
- To delete a custom style 271
- To delete a directory server instance 305
- To delete a disjoint searchbase 131
- To delete a password policy 327
- To delete a tab 112
- To delete a workflow 238
- To delete an administrator 47
- To delete an auxiliary object class 71
- To delete group members 150
- To delete reports 160
- To delete requests 235
- To deploy a style 270
- To disassociate a COREid Server and WebPass 319
- To enable ADSI for directory profiles 511
- To enable IWA on WebGate host 538
- To enable LDAP authentication for the Access Server 514, 521
- To enable or disable Lost Password Management 328
- To enable workflows 208
- To enter a password change redirect URL 331
- To expand a dynamic group 161
- To export workflows 238
- To find a workflow ticket 231
- To format and publish the report 159
- To generate a certificate request 346
- To generate a certificate with NetPoint CA 345
- To import photos to the directory 144
- To install a certificate 347
- To install the signed certificate 358
- To invoke a change attribute workflow 230
- To localize a panel 119
- To localize attribute display names 125
- To localize panel display names 124
- To localize reports 160
- To localize search result attributes 107
- To lock or unlock a ticket 235
- To log in to the Access System 36
- To log in to the COREid System 35
- To log out of the COREid System 41
- To manage a language 283
- To migrate transport security between Access Server and directory server 362
- To migrate transport security between COREid and directory server 361
- To modify a container limit 168
- To modify a COREid Server's connections to a WebPass 318
- To modify a WebPass 314
- To modify a workflow 238
- To modify a workflow panel 241
- To modify an authentication scheme scheme for a password policy 329
- To modify an object class type 67
- To modify attributes specific to the User, Group, or Organization Manager 125
- To modify password policy parameters 326
- To modify WebPass through a command line 315
- To monitor a workflow 233
- To order tabs in the Organization manager 112
- To preload the User, Group, and Organization Managers 229
- To process a ticket 231
- To re-configure the Access Server 301
- To reference photos in a file system 145
- To remove a WebPass 315
- To re-run Access Manager setup 300
- To re-run COREid System setup 299
- To revert to your own identity 56
- To run a workflow in the Group Manager 209
- To select attributes savailable for a step 204
- To select the class attribute 68
- To select what users see in My Groups and View Member Profiles 110
- To set or modify attribute permissions 138
- To set or modify logging and auditing policies 154
- To set the default style 272
- To set the globalParams.xml file 163
- To set the searchbase 128

- To set up a redirect URL 332
- To set up Access Manager for Active Directory 519
- To set up Access Server for Active Directory 519
- To set up additional directory server profiles 489
- To specify additional auxiliary object classes in User Manager 530
- To specify failover with Active Directory 520
- To specify NetPoint and Master Identity Administrators 46
- To specify transport security for Access Server 349
- To specify transport security for AccessGate 349
- To specify what attribute can be used in a search 106
- To subscribe to multiple groups 153
- To test IWA 540
- To test the workflow 209
- To turn auditing on and off 413, 415
- To update NetPoint configuration data 526
- To update the COREid Servers license key 280
- To use ANR in a search 528
- To use the Query Builder 132
- To use the Selector 40
- To verify ANR attribute access control 528
- To view a configured WebPass 311
- To view a directory server profile 298
- To view a panel 115
- To view an application-specific Modify Attribute page 80
- To view and add container limits 165
- To view and export a workflow summary 236
- To view configured object classes 65
- To view configured styles 267
- To view COREid Servers associated with a WebPass 318
- To view COREid System cache details 283
- To view current workflow panel settings 239
- To view group members 148
- To view Group Type panels 121
- To view language-specific workflow panel information 243
- To view logging and auditing policies 153
- To view or enter a COREid System license key 279
- To view or modify configured logs 390
- To view or modify COREid Server parameters 288
- To view or modify COREid Server settings 277
- To view or modify reports 160
- To view or modify tab configuration information 101
- To view password policies 322
- To view the Modify Attribute page from the System Console 80
- To view the Search Result attributes 106
- Process overview
 - During IWA authentication 537
 - Using a Create User workflow example 177
- profile panels
 - Organization Profile 116
 - User Profile 116 to 120
- profiles
 - role of object class attribute 66

- Provide Info And Approval workflow action 187
- Provide Info workflow action 187
- Provisioning 253
 - Object template file format 257
 - object templates, introduction 254
 - overview 253
 - task overview 254
 - Using template objects in a workflow 256
 - using workflows 254
- provisioning
 - and template objects 256
 - and the Identity Event API 264
 - and workflows 256
 - configuring object templates 257
 - how template objects are displayed 259
 - introduction 253
 - sending data to the back-end system 264
 - task overview 254
 - template vs. LDAP objects 254
- Provisioning external application accounts 253

Q

- Query Builder
 - building Advanced LDAP filters 135 to 137
 - building LDAP filters 132 to 133
- QuickStart tool, workflows
 - about 190
 - defining 191
 - Participant assignment, default 191

R

- redirect URL
 - about 326
 - entering 331
 - Password Change Redirect URL 330
 - Password Expiry Warning Redirect URL 332
 - setting up 332
- related documentation 18
- relevant_data workflow parameter 246
- replicating components
 - cloning and synchronizing 366
- replication frequency, Active Directory recommendation 500
- reporting features, contrasted and compared 371
- reports
 - disallowed data types 82
 - User Manager 156 to 159
- Request workflow action 187
- Requests page 230
- re-run Access Manager setup 300
- re-run Access Server setup 301
- re-run COREid System setup 299
- re-run NetPoint setup 299
- Resolving Ambiguous Names 523

- ResourceFilterSearchScope 199
- RFC 821 282
- rights
 - administrator rights, changing 137
 - delegated administrator rights 53
 - methods for assigning rights 54
 - setting or modifying by group 138
 - setting or modifying by user 138 to 141
 - simplified group access control, about 163
- rights, User Manager
 - about delegating 47
 - access rights, specifying 138 to 141
 - ASP delegated administration example 51
 - associating with attributes 142
 - delegated administrators, adding 52 to 54
 - extranet delegated administration example 49
 - intranet delegated administration example 50
- rules
 - about 83
 - compared to list 83
 - defining rules 83

S

- Screens
 - common elements 38
 - About link 39
 - application navigation bar 38
 - Logout 41
 - main body 38
 - online Help 38
 - side navigation bar 38
- search filter
 - ANR 524
- Search Result attributes
 - Organization Manager, enabling search functionality 106
 - User Manager attributes 106
- search results
 - object class attribute, role of 66
- searchbases
 - about 126
 - setting 128 to 130
- Searches
 - ANR 524
- searches
 - limiting 127
- searching
 - for a workflow ticket 230
- Secure Socket Layer. *See* SSL
- Select Groups workflow action 188
- self-registration
 - enabling 247
 - Self-Registration workflow action 188
- self-registration step 188
- Semantic types 74
- semantic types

- Challenge 328
 - determining for attributes 82
- Group Manager 82
- Organization Manager 82
- Response 328
- User Manager 82
- sending data to a back-end application 264
- sendMailFromEmail 188
- sendMailFromName 188
- session timeout, configuring 278 to 279
- Setting Up Integrated Windows Authentication 537
- Show dynamic groups display option 111
- Show dynamic user members of this group display option 111
- Show groups you are a member of display option 111
- Show groups you are an administrator of display option 111
- Show groups you are an owner of display option 111
- Show nested groups display option 111
- Show nested user members of this group display option 111
- Show static group display option 111
- Show static user members of this group display option 111
- Side navigation bar 38
- Siemens DirX 295
- signed certificate, installing 358
- silent mode
 - cloning and synchronizing 366
- Simple mode
 - See also* transport security certificate; transport security integration; transport security modes
 - generating a certificate 345
 - process overview 344
 - server definitions, updating 343
 - transport security mode password, changing 363
 - WebPass definitions, updating 343
- simple mode
 - migrating from simple or cert mode 356
- simplified access control
 - reserved words 165
- sample gscacparams.xml file 163
- single sign-on, integration
 - Active Directory requirements 495
- sizeLimit parameter 505, 509
- Smart Card Authentication 542
- SMTP message 188
- SMTP Server configuration function 281
- SSL
 - ADSI configuration requirement 501
- static audit reports 368
- statically-linked auxiliary classes 529
- structural object class
 - about 62, 529
 - available attributes, increasing 109
 - Group Type panel 122
- style sheets
 - creating and deploying 269 to ??

- Under Construction status 269
- styles, customized
 - about 266
 - Classic Style 271
 - configured styles, viewing 267
 - default style 272
 - deleting a style 271
 - new styles, creating and deploying 269 to ??
 - style name, changing 271
 - style sheets, about customizing 271
- Subflow Approval workflow action 188
- subflows
 - See also* workflows
 - advantages of 189
 - approval step 212
 - asynchronous operations, performing 228
 - calling from a workflow 211
 - Create User workflow example 189
 - creating (procedure) 210
 - creating (process overview) 210
 - filter criteria, importance of 190
 - status of 212
- substitution syntax, using with LDAP filters 89
- Supports MHTML email option 281
- synchronizing 366
- Synchronous Mail option 282

T

- tabs, localizing 104
- tabs, Organization Manager
 - adding 105
 - configuration conflicts, resolving 71
 - resequencing 112
- tabs, User Manager
 - class type 103
 - resolving configuration conflicts 81
 - role of 101
 - tab attribute reports, configuring 156 to 159
 - tab attribute reports, deleting 160
- target application
 - provisioning to 256
- Task overview
 - Configure ADSI for the Access System 513
 - configuring a provisioning solution for a back-end application 254
 - Configuring ADSI for the COREid System 510
 - Define a Create Location workflow 251
 - Delegating administrators 48
 - Displaying information on an application 100
 - Enable Location functionality and users 250
 - Enabling database auditing 416
 - Enabling Location functionality 146
 - Implementing Lost Password Management 328
 - Re-running system setup 299
 - Set up multiple COREid Servers 284
 - Setting Up Integrated Windows Authentication 538

- Setting up NetPoint for dynamic auxiliary classes 530
- Setting up NetPoint for the audit database 417
- Setting up NetPoint to use ANR during searches 525
- To install and configure MySQL 419
- Template objects 256
 - how they are used in workflows 256
 - See also, Provisioning
- template objects 256
 - and workflows 256
 - configuring 257
 - elements in the template 260
 - file format 257
 - how displayed in COREid 259
- Testing IWA Implementation 540
- Ticket information panel 240
- Ticket search table 240
- timeLimit parameter 505, 509
- timeouts
 - Active Directory 520 to 521
- To obtain the MySQL installation package 420
- To set up the MySQL user account 420
- transport request modes, changing 356
- transport security
 - Access Manager 361
 - Access Server 362
 - COREid Server 361
 - directory server 361
 - migrating modes 344
- transport security certificate
 - certificate request, generating 346, 356
 - internal CA, generating certificate 354
 - server definitions, updating 343
 - signed certificate, installing 347, 358
 - simple certificate password, changing 363
 - simple mode, generating a certificate 345
- transport security integration
 - AccessGate, updating definitions 349
 - changing modes, considerations 350
 - compatibility between COREid Servers and WebPass instances 313
 - updating definitions 349
 - WebGate, updating definitions 349
 - WebPass definitions, updating 343
- transport security modes
 - configuring 337
 - Simple mode, certificates 345 to 363
 - Simple mode, updating server definitions 343
- typographical conventions 19

U

- Updating NetPoint Configuration Data for ANR 526
- useGCForAuthn parameter 505, 509
- useImplicitBind flag 502, 507
- useImplicitBind parameter
 - Access System, values 509
 - COREid System, values 505

- useLDAPBind parameter 509
- user accounts
 - provisioning 253
- user domain, determining for Active Directory forest configuration 496
- user feedback
 - email configuration 280
- User Manager
 - See also* Group Manager; Organization Manager
 - data types 82
 - display types 83
 - mouse over message 103
 - reports 156 to 159
 - searchbase 126 to 130
 - self-registration, enabling 247
 - Semantic Types, listed 82
 - tab image 103
 - tabs 101 to 103
- User Manager attributes
 - additional attributes, configuring 80
 - additional attributes, modifying 82
 - assignment methods 140
 - configuration keyboard shortcuts 141
 - conflicting attribute configurations, resolving 81
 - derived attributes, assigning to a tab 96
 - evaluation order 142
 - multiple values, use of 82
 - Search Result attributes, modifying 106
 - Search Result attributes, selecting 106
 - trustee (participant) access 139
- User Manager workflows
 - actions, table of 186 to 188
 - first step, valid action 202
 - workflow types 179
- User Profile
 - configuring header panel 113, 114
- User Profile panels
 - adding 116 to 118
 - modifying, deleting, or ordering 120
- User Profiles
 - photos in profiles 143
- useraction workflow parameter 246
- userPrincipleName 503
- users
 - adding 179
 - deactivating 232
 - reactivating 232
- Using Access System Password Management 540
- Using ANR in NetPoint Searches 528

V

- validate_password plug-in
 - value 329
- Verifying ANR Attribute Access Control 527

W

- Web master feedback, configuring 280
- WebGate
 - transport security, updating definitions 349
- WebPass
 - failover actions 313
 - transport security modes 313, 343
 - viewing Identity Servers associated with 318
- WebPass instances
 - adding or restoring 312 to 314
 - associating with a COREid Server 319
 - COREid Server, disassociating 319
 - deleting 315
 - modifying 314
 - uninstalling compared to deleting 315
 - viewing 311
- WhichAttrIsLogin parameter 495
- workflow
 - static participants 214
- workflow actions
 - See also* workflow steps
 - about 194
 - about defining 182, 194
 - basic actions 194
 - table of 186 to 188
- Workflow monitor table 240
- workflow panels 239 to 241
- workflow parameters 246
- Workflow profile panel 240
- workflow QuickStart tool
 - Participant assignment, default 191
 - workflows, defining 191
- workflow steps
 - about defining 180
 - asynchronous operations, performing 228
 - attribute properties, configuring 205 to 206
 - attributes, selecting 204
 - custom actions, about 244
 - external actions, about 245
 - first action, defining 202
 - pre and post actions, about 244
 - subflow, invoking 211
 - subsequent actions, defining 207
 - termination of 181
- Workflow steps profile panel 240
- Workflows
 - and template objects 257
- workflows
 - about testing 209
 - adding roles 246
 - archiving 234
 - controlling performance of 243
 - copying 237
 - create location 250
 - Create User workflow, sample 171
 - creating new workflow 196 to 198
 - deleting 238

- enabling 208
- end use of 229
- export parameter settings 235, 238
- Group Manager workflow types 179
- Group Type designation 195
- Group Type panel, role in running workflow 209
- invoking 229
- modifying 237
- monitoring 233
- Organization Manager, workflow types 180
- pending 230
- purging 234
- saving, restriction on 205
- targets, defining 200

- terminology 201
- tickets 230
- User Manager workflow types 179
- using 229
- Workflow Definition applet, accessing 195
- workflow domain, role in creation process 197
- workflow summary, viewing 235
- workflow types by application 179

X

- XML style sheets, creating 269 to ??