

# **Oracle® COREid Access and Identity**

# **Integration Guide**

**10g Release 2 (10.1.2)  
Part No. B19014-01**

**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](mailto: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

	Intended Audience .....	19
	COREid Documentation.....	20
	Typographical Conventions .....	21
	Contact Information.....	21
	Corporate Headquarters .....	21
	Before Contacting Customer Care .....	21
	Accessing the Customer Care Knowledge Base .....	22
<b>Chapter 1</b>	<b>Introduction.....</b>	<b>23</b>
	About NetPoint 7.0 Integrations.....	23
<b>Chapter 2</b>	<b>Integrating NetPoint Ready Realm for BEA .....</b>	<b>27</b>
	About NetPoint Ready Realm for BEA.....	28
	NetPoint Components .....	28
	BEA WebLogic Components .....	29
	Integration Architecture.....	30
	BEA WebLogic and NetPoint Objects .....	31
	Supported Versions and Platforms .....	34
	Configuring NetPoint Ready Realm for BEA.....	34
	Preparing for Integration with NetPoint Ready Realm .....	35
	Adding an AccessGate .....	35
	Generating NetPoint Ready Realm for BEA Information .....	37
	Installing NetPoint Ready Realm for BEA.....	44
	Preparing for Ready Realm Installation .....	45
	Installing NetPoint Ready Realm for BEA .....	46
	Configuring WebLogic for Ready Realm.....	52
	Appending the NetPointBEARealm.properties File .....	53
	Configuring NetPoint Ready Realm for WebLogic 6.x .....	54
	Configuring NetPoint Ready Realm for WebLogic 7.x .....	59
	Modifying WebLogic Server and Portal Startup Scripts .....	63
	Protecting Resources With Policy Domains.....	65
	Example: Defining a J2EE_Role to protect an EJB or a WebApp .....	66

Example: Defining a J2EE_Acl to protect an EJB or a WebApp .....	70
Advanced Configuration Options .....	73
Changing weblogic_system User .....	74
Configuring Single Sign-On for the WebLogic Portal .....	74
Changes to Workflow .....	79
Samples for NetPoint Ready Realm for BEA .....	81
Single Sign-On (SSO) with Form Login .....	81
Roles in Deployment Descriptor .....	83
Single Sign-On (SSO) Servlet .....	84
Mapping EJB Roles to NetPoint Roles .....	85
WebLogic Dynamic Role Mapping .....	86
Custom ACL .....	87
Implementation Notes for Active Directory .....	88
NetPoint Ready Realm for BEA Configuration Fails .....	88
Create User Fails Talking to Active Directory Forest .....	89
Set Active Directory in NetPointBEARealm.properties .....	89
Set the User Mapping in NetPointBEARealm.properties .....	89
Set Authentication Scheme for Active Directory .....	90
Troubleshooting .....	90

<b>Chapter 3</b>	<b>Integrating NetPoint Security Provider for WebLogic.....</b>	<b>101</b>
	About the NetPoint Security Provider .....	102
	WebLogic and NetPoint Integration Points .....	102
	Differences From NetPoint Ready Realm for BEA .....	103
	Integration Architecture .....	104
	Authentication for Mixed Web and Non-Web Resources .....	105
	Authentication for Web-Only Resources .....	106
	Authentication for the Portal .....	108
	Supported Versions and Platforms .....	110
	Online Assistance .....	111
	Installing and Configuring the Security Provider.....	112
	Preparing the Environment .....	112
	Installing NetPoint Security Provider for WebLogic .....	113
	Completing a Typical Installation .....	114
	Completing Advanced Installation .....	115
	Setting Up WebLogic Policies in NetPoint .....	117
	Running the NetPoint Policy Deployer .....	122
	Manually Configuring WebLogic Policies in NetPoint .....	125

Mapping WebLogic Resources to NetPoint Resources .....	131
Preparing the WebLogic Environment .....	135
Configuring the COREid Server .....	140
Configuring Multiple WebPass Instances .....	141
Configuring SSO for the Portal Server .....	141
Configuring web.xml .....	142
Configuring the login.jsp used by the Login Portlet .....	142
Copying ObLoginFilter.class in the WEB_INF/classes .....	144
Completing Setup .....	144
Testing SSO for the Portal Server .....	145
Context-Specific Authorization .....	146
Audit Files .....	148
Debug Log Files .....	148
User Creation/Deletion and Group Creation .....	149
Configuration Files .....	151
NetPointProvidersConfig.properties .....	152
NetPointWeblogicTools.properties .....	160
Implementation Notes for Active Directory .....	161
Configuring NetPoint Connector for WebLogic .....	161
Setting a Domain in NetPointProvidersConfig.properties .....	162
Tips .....	165
WebLogic Portal Admin Console Changes .....	167
References .....	169
Troubleshooting NetPoint Security Provider for WebLogic .....	170
Additional Resources .....	176

## **Chapter 4      Integrating NetPoint with IBM WebSphere ..... 179**

About the NetPoint Connector for WebSphere .....	180
WebSphere Components .....	182
NetPoint Connector for WebSphere Components .....	183
Integration Architecture .....	184
Scenario 1: Use of NetPointWASRegistry .....	184
Scenario 2: Architecture for Single Sign-On .....	186
Mapping NetPoint Users and Groups to Security Roles in WAS .....	188
Supported Versions and Platforms .....	189
Preparing to Install the NetPoint Connector .....	190
Preparing Your Environment .....	191

Configuring the COREid System for WAS Integration .....	192
Configuring the Access System for WAS Integration .....	195
Configuring Resource Protection in NetPoint .....	198
Defining a Policy Domain for the WebSphere v6.0 Administration Console ...	204
Installing the NetPoint Connector for WebSphere .....	206
Launching the Installation .....	206
Defining the Installation Directory .....	207
Specifying Connector Details .....	207
Completing Details for the WebGate .....	209
Specifying AccessGate Details .....	212
Installing a Certificate .....	215
Configuring Multiple WebPass Instances for the NetPoint Connector .....	216
Completing NetPoint Connector Setup.....	217
Setting Up the NetPoint Connector for WebSphere .....	217
Testing Environment Setup .....	219
Configuring WebSphere Application Server v4 .....	221
Enabling the NetPointWASRegistry in WAS 4 .....	221
Testing NetPointRegistry for WAS v4, v5, and v6 .....	224
Configuring the TAI for WebSphere 4 and NetPoint .....	226
Enabling Logging for TAI for WAS 4 .....	229
Testing NetPoint Single Sign-on for WAS v4.x .....	231
Integrating NetPoint with the Portal v4 .....	235
WebSphere Portal Component .....	235
NetPoint Custom Member Repository .....	235
Integration Scenario with the NetPoint CMR .....	237
Setting up the WebSphere Portal for use with NetPoint .....	239
Managing Users and Groups .....	244
Password Management .....	244
Access Control for the WebSphere Portal .....	245
Configuring Single Sign-On Functions for the Portal .....	245
Configuring WebSphere Application Server v5 .....	247
Enabling the NetPointWASRegistry in WAS v5 .....	247
Testing the NetPointWASRegistry for WebSphere v5 .....	251
Configuring the TAI for WebSphere v5 .....	251
Integrating NetPoint and WebSphere Portal v5.....	255
About Integration with the NetPoint CMR .....	259
Setting up the WebSphere Portal v5.0.2 for NetPoint .....	260
Setting Up NetPoint with WebSphere Portal v5.1 .....	265
Managing Users and Groups with Portal v5 .....	273

Password Management with Portal v5 .....	274
Access Control for the WebSphere Portal v5 .....	274
Configuring Single Sign-on Functions for the Portal v5 .....	275
Configuring the WebSphere Application Server v6.....	276
Enabling the NetPointWASRegistry for WAS v6 .....	277
Testing the NetPointWASRegistry for WebSphere v6 .....	278
Configuring the TAI for WebSphere v6 .....	279
Testing the TAI for WAS v6 .....	282
Enabling Logging for TAI for WAS v6 .....	283
Configuration Files .....	283
NetPointWASRegistry.properties .....	283
WebGate.properties Configuration File .....	290
TrustedServers Configuration File .....	292
Implementation Notes for the TAI .....	292
Implementation Notes for Active Directory.....	294
Configuring NetPoint Connector for WebSphere for Active Directory Forest ..	294
Set Active Directory Domain in NetPointWASRegistry.properties .....	295
Troubleshooting the NetPoint Connector for WebSphere.....	295
Troubleshooting the NetPoint Connector for Portal Server v5 .....	306

## **Chapter 5      Integrating Oracle Application Servers with NetPoint SSO..... 313**

About the NetPoint and Oracle Integration .....	314
NetPoint Features for Oracle Application Servers .....	314
Oracle9iAS and OracleAS 10g Infrastructure .....	314
Integration Architecture.....	315
Supported Versions and Platforms .....	316
Preparing Your Environment.....	317
Integrating NetPoint with Oracle9iAS.....	318
Configuring Oracle9iAS for Integration with NetPoint .....	319
Configuring NetPoint for Integration with Oracle9iAS .....	320
Configuring Logout .....	321
Testing NetPoint Integration with Oracle .....	322
Integrating NetPoint with OracleAS 10g .....	323
Configuring OracleAS 10g for Integration with NetPoint .....	323
Configuring NetPoint for Integration with OracleAS 10g .....	330
Protecting the Single-Sign On Login URL .....	331
Sample Files .....	332
Oracle9iAS Files .....	332

OracleAS 10g Files .....	336
Troubleshooting the Oracle Integration .....	341
Troubleshooting the Oracle9iAS Integration .....	341
Troubleshooting the OracleAS 10g Integration .....	344
<b>Chapter 6</b>	
<b>Integrating Plumtree Corporate Portal .....</b>	<b>349</b>
About the Integration with Plumtree Portal .....	349
Supported Versions and Platforms .....	352
Enabling Single Sign-On in Plumtree 4.5 .....	352
Creating an SSO Authentication Source .....	352
Creating an LDAP Authentication Source .....	354
Editing Configuration Files to Support SSO .....	359
Synchronizing LDAP Data with Plumtree Database .....	363
Setting Up COREid to Protect Plumtree 4.5 .....	367
Installing COREid Components .....	367
Creating a Policy Domain .....	368
Enabling Single Sign-on in PlumTree 5.0.4 .....	370
Creating an SSO Authentication Source .....	371
Creating an LDAP Authentication Source .....	372
Editing Configuration Files to Support SSO .....	373
Synchronizing LDAP Data with Plumtree Database .....	374
Enabling SSO Logout .....	375
Setting Up COREid to Protect Plumtree 5.0.4 .....	376
Creating a Policy Domain .....	377
Configuring the COREid WebGate .....	377
Configuring COREid WebGate for IIS .....	378
Integrating Other COREid Features .....	378
Using Gadgets to Create Integration Features .....	378
Enabling Anonymous Users to View Portal Guest Pages .....	385
Personalizing User Pages .....	387
Embedding COREid Management Functions .....	389
Importing COREid Dynamic Groups .....	393
Password Management .....	396
Self-Registration .....	396
Using the Knowledge Directory .....	397
Setting Knowledge Directory Preferences .....	397
Creating Folders .....	397
Uploading Documents .....	398

<b>Chapter 7</b>	<b>Integrating Siebel 7 with NetPoint SSO .....</b>	<b>399</b>
	About the Integration with Siebel 7 .....	399
	Siebel 7 Components .....	400
	Integration Architecture.....	401
	Supported Version and Platforms .....	403
	Preparing Your Environment.....	404
	Setting Up NetPoint SSO for Siebel Application Server .....	404
	Setting Up Siebel 7 for integration with NetPoint .....	404
	Setting up NetPoint for Integration with Siebel 7 .....	409
	Testing Integration Between NetPoint and Siebel .....	410
<b>Chapter 8</b>	<b>Integrating mySAP Applications with NetPoint SSO.....</b>	<b>413</b>
	About Integrating NetPoint with mySAP .....	413
	SAP Components .....	414
	SAP Internet Transaction Server .....	414
	Pluggable Authentication Service .....	415
	Integration Architecture.....	415
	Supported Versions and Platforms .....	417
	Preparing to Integrate NetPoint with SAP .....	417
	Setting up NetPoint SSO for mySAP .....	418
	Setting Up SAP for Integration with NetPoint .....	419
	Setting Up NetPoint for Integration with SAP .....	420
	Testing Integration Between NetPoint and SAP .....	421
	Integrating the SAP Enterprise Portal .....	422
	Preparing for SAP Portal Integration .....	423
	Integrating NetPoint with the SAP Portal .....	423
	Testing Integration Between NetPoint and SAP Portal .....	425
	Configuration Files .....	426
	Configuration Examples.....	426
	WGate.config .....	427
	Oblix.srvc .....	428
	global.srvc .....	428
	Template Examples .....	429
	Login.html .....	429
	extautherror.html .....	431
	redirect.html .....	431

## Chapter 9

<b>Integrating the RSA SecurID Authentication Plug-In .....</b>	<b>433</b>
About NetPoint and SecurID Authentication .....	433
RSA Components .....	434
Oblix Components .....	435
Integration Summary .....	437
Supported Versions, Platforms, and Requirements .....	437
RSA ACE/Server Requirements .....	438
NetPoint Access Server and ACE/Agent Requirements .....	440
NetPoint WebGate Requirements .....	441
Associate Portal Requirements .....	443
SecurID Authentication Scenarios .....	444
SecurID Authentication Sequence .....	444
Next Tokencode Sequence .....	446
New PIN Sequence .....	446
Integrating SecurID Authentication .....	448
Preparing Your Environment .....	449
Setting up the Access Server as an ACE/Agent .....	450
Setting Up a SecurID WebGate .....	454
Creating a SecurID Authentication Scheme .....	458
Protecting SecurID Resources .....	464
Testing the Policy Domain .....	468
Adding ACE/Server Users to NetPoint .....	468
NetPoint Authentication Plug-In Parameters .....	469
SecurID Plug-In Parameters .....	469
Credential Mapping Plug-In Parameters .....	471
Active Directory Forest Considerations .....	472
Prerequisites .....	472
Integrating SecurID with an Active Directory Forest .....	473
Troubleshooting .....	476
ACE/Agent Issues .....	476
ACE/Server Configuration File .....	477
CGI Directory on SecurID WebGates .....	477
Environment Variable on Unix Systems .....	477
Form-Based Authentication .....	477
Access Server Log .....	478
Web Server Logs .....	478
RSA ACE/Server Logs .....	478
Permissions .....	479
SecurID Plug-In Parameters with Modified HTML Fields .....	479

<b>Chapter 10</b>	<b>Integrating NetPoint with Smart Card Authentication.....</b>	<b>481</b>
	About Smart Card Authentication .....	481
	About NetPoint Components .....	482
	Integration Architecture.....	483
	Supported Versions and Platforms .....	485
	Setting Up Smart Card Authentication .....	485
	Preparing the Active Directory .....	485
	Preparing the CA and Enrolling for a Certificate .....	486
	Preparing IIS Web Servers .....	487
	Preparing NetPoint for Smart Card Authentication .....	487
	Protecting Resources with NetPoint .....	487
	Setting Up the IIS Manager .....	490
	About Policy Domains for Smart Card Authentication .....	490
	Client Certificate Authentication Schemes.....	491
	Smart Card Challenge Method, Parameter, SSL .....	492
	Plug-Ins for Smart Card Authentication .....	492
	Troubleshooting .....	494
	Problem Requesting X.509 Certificates .....	494
	Additional Resources .....	494
<b>Chapter 11</b>	<b>Integrating NetPoint with .NET Passport.....</b>	<b>497</b>
	About NetPoint and .NET Passport .....	497
	Supported Platforms, Versions, and Requirements.....	499
	Required Passport Components .....	499
	Required NetPoint Components .....	500
	Integration Architecture.....	500
	Integrating NetPoint and the Passport Plug-In.....	501
	Preparing for the Integration .....	502
	Installing DotNetGate on IIS .....	502
	Configuring NetPoint .....	503
<b>Chapter 12</b>	<b>Integrating NetPoint with Authorization Manager Services.....</b>	<b>507</b>
	About NetPoint and the AzMan Plug-In .....	507
	Authorization with the NetPoint AzMan Plug-In .....	509
	NetPoint Components and Requirements .....	512
	NetPoint Authorization Rules and Schemes .....	512

About the Authorization Manager .....	516
Authorization Stores .....	516
Applications and Scopes .....	517
Operations and Tasks .....	518
Roles .....	518
Groups .....	519
Rules .....	519
Auditing .....	520
Authorization Manager (AzMan) API .....	520
Examples .....	521
Example 1: An Expense Application .....	521
Example 2: NetPoint Configuration .....	525
Example 3: NetPoint AzMan Plug-In Authorization Process Flow .....	529
Configuring the NetPoint AzMan Plug-In .....	531
Preparing Your Environment .....	532
Creating an Authorization Scheme for the AzMan Plug-In .....	532
Protecting Resources .....	533
Defining Authorization Rules and Policies .....	534
Using the NetPoint AzMan Plug-In with the Access Server API .....	536
Troubleshooting .....	538

## **Chapter 13 Integrating the NetPoint Security Connector for ASP.NET ..... 539**

About ASP.NET .....	539
Security Principals and Security Identifiers (SIDs) .....	540
IPrincipal.IsInRole Method Syntax .....	540
About the Security Connector for ASP.NET .....	542
NetPoint Components and Requirements .....	542
The OblixHttpModule .....	543
The OblixPrincipal Object .....	543
Authorization with the Security Connector for ASP.NET .....	543
Using the Security Connector for ASP.NET .....	544
Setting Up Your Environment .....	545
Setting Up the ASP.NET Application for the NetPoint Security Connector .....	546
Setting Up the NetPoint Role Action .....	548

	NetPoint Role-Based Authorization .....	549
<b>Chapter 14</b>	<b>Integrating Workflows With MIIS Provisioning .....</b>	<b>551</b>
	About Provisioning with MIIS .....	552
	Integration Architecture .....	553
	About Integrating NetPoint and MIIS .....	555
	Preparing Your Environment.....	556
	Preparing for Oblix Management Agent Installation .....	557
	Preparing for Password Synchronization .....	557
	Installing the Oblix Management Agent .....	558
	Confirming Installation .....	559
	Configuring the Oblix Management Agent .....	562
	Creating a Management Agent in MIIS .....	563
	Configuring the SQL Server Connection .....	563
	Configuring the MIIS Database Columns .....	564
	Configuring Join and Projection Rules .....	565
	Configuring the Attribute Flow .....	567
	Configuring Deprovisioning .....	569
	Configuring Extensions .....	569
	Configuring Run Profiles for the Agent .....	569
	Copying the MIIS Files to COREid.....	570
	Configuring the Database, COREid, and SSO .....	571
	Configuring the SQL Database .....	572
	Configuring a COREid Workflow for MIIS Provisioning .....	573
	Configuring Single Sign-On .....	576
	Customize the Management Agent Configuration File .....	577
	Configuring Logging.....	580
	Logging Parameters .....	580
	About the ObMA Application.....	581
	Command-Line Parameters .....	581
	Examples .....	582
	Using a COREid Workflow to Delete Identities from Metaverse .....	582
	.....	583
<b>Chapter 15</b>	<b>Integrating NetPoint SSO with SharePoint Portal Server .....</b>	<b>585</b>
	About NetPoint and the SharePoint Portal Server .....	585
	About Windows Impersonation .....	586

Supported Platforms and Requirements .....	586
Required Microsoft Components .....	586
Required NetPoint Components .....	588
Request Processing by the SPPS Integration .....	588
Integrating NetPoint with SPPS .....	589
Installing Microsoft Components .....	589
Installing NetPoint Components .....	593
Setting Up Impersonation .....	595
Creating a Trusted User Account .....	596
Assigning Rights to the Trusted User .....	597
Binding the Trusted User to Your WebGate .....	598
Adding an Impersonation Action to a Policy Domain .....	599
Adding an Impersonation dll to IIS .....	600
Testing Impersonation .....	601
Completing NetPoint to SPPS Integration .....	604
Configuring IIS Security .....	604
Configuring the Wildcard Extension .....	605
Editing web.config .....	607
Uploading User Data .....	608
Testing Your Integration .....	608

## **Chapter 16      Integrating NetPoint and the Content Management Server..... 611**

About NetPoint and the MCMS .....	611
About Windows Impersonation .....	612
Supported Platforms and Requirements .....	612
Required NetPoint Components .....	612
Required Microsoft Components .....	613
Request Processing by the Integration.....	613
Integrating NetPoint with the MCMS .....	614
Installing NetPoint .....	615
Installing Microsoft Components .....	615
Integrating NetPoint with the MCMS .....	616
Setting Up Impersonation .....	616
Completing the NetPoint to MCMS Integration .....	617
Testing the NetPoint to MCMS Integration .....	618

<b>Chapter 17</b>	<b>Integrating NetPoint with OctetString VDE .....</b>	<b>621</b>
	About NetPoint and VDE Integration .....	622
	Key Terms and Features .....	622
	Federated Data Stores .....	625
	Split Profiles .....	627
	Aggregated Namespaces .....	628
	Aggregated Schema Mapping .....	629
	Integration Limitations .....	630
	About Limitations on Multi-Value Attributes .....	630
	About Limitations on Embedded Virtual Data Sources .....	632
	Integration Architecture .....	632
	About VDE Drivers and Adapters .....	634
	About NetPoint-Specific Data .....	634
	About Schema Extension .....	635
	Virtual Directory Schema .....	637
	Target Directory Schemas .....	637
	About Adding Attributes to Target Database Tables .....	638
	Customer Schemas .....	638
	Integration Scenarios and Limitations .....	639
	Heterogeneous LDAP Directories .....	640
	Multiple RDBMS Databases .....	641
	Split-Profiles .....	643
	Integration Requirements .....	644
	Security Connection Support .....	646
	Authentication Support .....	647
	Access Control Support .....	647
	Failover Support .....	648
	About the NetPoint-VDE Integration Process .....	649
	Integrating VDE when Installing NetPoint .....	649
	Integrating VDE with Existing NetPoint Installations .....	650
	Preparing Your Environment .....	652
	Identifying Factors for Designing Your Integration .....	652
	Preparing Directory Servers for Integration .....	655
	Preparing Relational Databases for Integration .....	656
	Installing and Configuring VDE and DME .....	657
	Installing VDE .....	657
	Installing DME .....	658
	Creating a Project Space and Server .....	658
	Obtaining/Updating Sample Adapter and Mapping Templates .....	659

Deploying JDBC Driver Libraries for Your RDBMS .....	660
Configuring the VDE SSL Listener (Optional) .....	661
Installing the First NetPoint COREid Server .....	663
Extending Directory Schemas .....	665
Creating Mapping Files for Adapters .....	667
Creating Data Store Adapters .....	669
Creating Adapters for LDAP Directories .....	669
Configuring a Database Adapter .....	673
Creating a Split-Profile Adapter .....	675
Creating a Multiple-Directories Adapter .....	677
Customizing Adapters and Mapping Files .....	679
Customization Examples .....	680
Customizing General Settings for NetPoint .....	694
Customizing Routing Settings .....	696
Editing an Adapter Plug-in to Refer to Your Mapping File .....	696
Completing NetPoint Installation and Setup .....	698
Testing Your Integration .....	699
Reference Information .....	700
NetPoint Auxiliary Attributes .....	700
About the Data Anywhere Toolkit .....	703
NetPoint-VDE Integration Templates .....	708
Templates for Active Directory .....	708
Templates for ADAM .....	711
Templates for Sun Directory Server .....	714
Templates for eDirectory .....	715
Database Template: OblixDBAdapterUsingScript .....	716
Schema Mapping Script Templates .....	716
Integration Tips .....	717
Database Connectivity Tips .....	719
Troubleshooting .....	721
Directory Server Problems .....	721
Multi-Value Attribute Problems .....	721
Secondary Data Store Problems .....	722
Unexpected Group Deletion Problem .....	723

# Preface

The *COREid Integration Guide* provides information about integrating COREid with third-party applications servers and portals.

---

**Note:** Oracle *COREid* was previously known as *Obliv Netpoint*. All legacy references to Obliv and NetPoint, for example, in screen shots, illustrations, and documentation titles, should be understood to refer to Oracle and COREid, respectively. Some integrations operate with earlier versions of NetPoint.

---

This Preface covers the following topics:

- “Intended Audience” on page 19
- “COREid Documentation” on page 20
- “Typographical Conventions” on page 21
- “Contact Information” on page 21

## Intended Audience

This guide is intended for administrators who are responsible for integrating their product with COREid.

This document assumes that you are familiar with your LDAP directory and Web servers, as well as the product you are integrating.

# 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.

# 1 Introduction

This chapter provides an overview of the NetPoint 7.0 integrations that are described in this guide. For an introduction to NetPoint, see the *Introduction to NetPoint 7.0 Guide*.

---

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

---

## About NetPoint 7.0 Integrations

Integrating NetPoint 7.0 with other applications and portals requires some knowledge of both products. This guide provides the details you need to successfully set up NetPoint and the applications and portals you will integrate with NetPoint.

---

**Note:** Pay close attention to the specific requirements for your integration.

---

The following integrations are described in this guide:

- **BEA WebLogic**—The NetPoint 7.0 Ready Realm for BEA provides identity management, access control, and single sign-on across J2EE applications developed on the BEA WebLogic platform. For more information, see “Integrating NetPoint Ready Realm for BEA” on page 27 and “Integrating NetPoint Security Provider for WebLogic” on page 101.
- **BEA WebLogic Security Service Provider Interface (SSPI)**—The NetPoint 7.0 Security Provider for WebLogic ensures that only appropriate users and groups can access NetPoint-protected WebLogic resources to perform specific operations. The NetPoint Security Provider also allows you to configure single sign-on between NetPoint and WebLogic resources.

---

**Note:** The NetPoint Ready Realm for BEA supports WebLogic running in compatibility mode. See “Integrating NetPoint Ready Realm for BEA” on page 27 for details.

---

- **IBM WebSphere**—The NetPoint 7.0 Connector for WebSphere provides identity management, access control, and single sign-on across J2EE resources

and applications developed on the IBM WebSphere platform. For more information, see “Integrating NetPoint with IBM WebSphere” on page 179.

- **Oracle Application Servers**—Integrating NetPoint with the Oracle*9i*AS and OracleAS 10g Infrastructure allows the use of NetPoint single sign-on (SSO) and identity management functionality across Web-based applications including Oracle applications, Oracle eBusiness Suite, and other NetPoint-protected enterprise resources and applications. For more information, see “Integrating Oracle Application Servers with NetPoint SSO” on page 313.
- **Plumtree Portal**—NetPoint provides identity management, access control, and single sign-on for the Plumtree Corporate Portal. “Integrating Plumtree Corporate Portal” on page 349.
- **Siebel 7 e-business Platform**—Siebel 7 is a Web-based suite that combines customer relationship management, partner relationship management and employee relationship management applications. For more information, see “Integrating Siebel 7 with NetPoint SSO” on page 399.
- **mySAP**—Integrating NetPoint with mySAP allows the use of NetPoint functionality across all mySAP Web-based applications and other NetPoint-protected enterprise resources and applications. For more information, see “Integrating mySAP Applications with NetPoint SSO” on page 413.
- **RSA SecurID Authentication**—NetPoint 7.0 supports RSA Security features and provides the SecurID authentication plug-in and components needed to integrate a native SecurID authentication scheme into NetPoint policy domains for Web single sign-on. For more information, see “Integrating the RSA SecurID Authentication Plug-In” on page 433.
- **OctetString Virtual Directory Engine (VDE)**— NetPoint 7.0.2 supports integration with the OctetString VDE, which combines user data from multiple data sources to create an aggregated virtual directory. To a NetPoint user, this aggregated virtual directory looks and behaves just like any other LDAP directory, and is compatible with the full range of NetPoint features. For more information, see “Integrating NetPoint with OctetString VDE” on page 621.
- **Microsoft Products**—NetPoint 7.0 supports integration with the following Microsoft features and services:
  - **Smart Card Authentication**—NetPoint 7.0 supports smart card authentication with Active Directory and IIS Web servers using ActivCard<sup>®</sup> Cryptographic Service Provider (CSP) for Windows 2000, ActivCard Gold utilities, and ActivCard USB Reader v2.0 in homogeneous Windows environments. For more information, see “Integrating NetPoint with Smart Card Authentication” on page 481.

- **Passport Authentication**—NetPoint 7.0 includes a custom authentication ISAPI plug-in, DotNetGate, that passes Passport authentication information to WebGate for use in a NetPoint external authentication scheme, which maps Passport users to NetPoint users. For more information, see “Integrating NetPoint with .NET Passport” on page 497.
- **Authorization Manager**—NetPoint includes a custom authorization plug-in, the NetPoint AzMan Plug-in to use Authorization Manager services to make authorization decisions for Access Server clients, including WebGates and callers of the Access Server API. For more information, see “Integrating NetPoint with Authorization Manager Services” on page 507.
- **ASP.NET**—The NetPoint 7.0 Security Connector for ASP.NET can convert authorization actions into roles. For more information, see “Integrating the NetPoint Security Connector for ASP.NET” on page 539.
- **Microsoft Identity Information Server (MIIS) Provisioning**—NetPoint 7.0 enables you to use the COREid System workflow functionality to add, modify, and delete information about users and to propagate this information to different target data sources that are provisioned using MIIS. For example, you can configure an Add or Modify User workflow in COREid to provision user accounts in Exchange. For more information, see “Integrating Workflows With MIIS Provisioning” on page 551.
- **Microsoft SharePoint Portal Server (SPS) 2003**—NetPoint 7.0 provides authentication for SPS resources and services, URL level authorization, and single sign-on for seamless navigation between the portal and other protected resources. The SharePoint Portal Server will enforce application-specific authorization policies for capabilities within the SharePoint application and offers the option of controlling access to specific documents. For more information, see “Integrating NetPoint SSO with SharePoint Portal Server” on page 585.
- **Microsoft Content Management Server (CMS) 2002**—NetPoint 7.0 provides authentication for CMS resources and services, URL level authorization, and single sign-on. Application level authorization will be performed by the CMS. NetPoint will pass the appropriate role to the CMS. The CMS will enforce its own roles. For more information, see “Integrating NetPoint and the Content Management Server” on page 611.



# 2 Integrating NetPoint Ready Realm for BEA

This chapter describes the integration of NetPoint 7.0 with the BEA WebLogic platform running as a custom realm implementation.

This chapter covers the following topics:

- “About NetPoint Ready Realm for BEA” on page 28
- “Integration Architecture” on page 30
- “Supported Versions and Platforms” on page 34
- “Configuring NetPoint Ready Realm for BEA” on page 34
- “Installing NetPoint Ready Realm for BEA” on page 44
- “Configuring WebLogic for Ready Realm” on page 52
- “Protecting Resources With Policy Domains” on page 65
- “Advanced Configuration Options” on page 73
- “Samples for NetPoint Ready Realm for BEA” on page 81
- “Implementation Notes for Active Directory” on page 88
- “Troubleshooting” on page 90

# About NetPoint Ready Realm for BEA

NetPoint 7.0 Ready Realm for BEA provides centralized access control and transparent single sign-on across J2EE applications developed on the BEA WebLogic platform.

The NetPoint Ready Realm for BEA is a NetPoint 7.0 AccessGate that is also a WebLogic Custom Security Realm. It establishes a native connection between the BEA WebLogic Server and Oblix NetPoint, providing an easy way for BEA WebLogic customers to leverage the NetPoint unified security and Web access management framework to control user access and identity for their business applications. With the NetPoint BEA Realm, you can use authentication, authorization, access control, single sign-on (SSO), delegated administration, dynamic groups, workflows, identity management, and all other NetPoint features to support your Web servers, applications servers, and legacy platforms.

This section describes the functionality of NetPoint Ready Realm for BEA, including the components required for the integration.

## NetPoint Components

In addition to the NetPoint components described in the *Introduction to NetPoint 7.0* guide, the following NetPoint Components are used in integration with BEA WebLogic:

- “Access Manager API” on page 28
- “Access Server SDK” on page 29
- “IdentityXML” on page 29

## Access Manager API

The Access Manager API maps the definition of WebLogic groups and roles to J2EE roles in the NetPoint Access Manager, where they can be defined by access policies. See “Configuring NetPoint Ready Realm for BEA” on page 34 for more information.

---

**Note:** The Access Manager API is packaged with the Access Server SDK under one shared library when you install the NetPoint Ready Realm for BEA package.

---

## Access Server SDK

The Access Server API is used to implement authentication and authorization for BEA WebLogic resources. The Access Server SDK is responsible for managing and enforcing access control by ACLs.

## IdentityXML

IdentityXML is used to create and delete users from the WebLogic Portal Admin tool using NetPoint workflows. An IdentityXML request is made from NetPoint Ready Realm for BEA to WebPass using a Simple Object Access Protocol (SOAP) request over HTTP/HTTPS.

## BEA WebLogic Components

The following BEA WebLogic components are required for integration with NetPoint.

- “BEA WebLogic Server” on page 29
- “BEA WebLogic Portal” on page 30

## BEA WebLogic Server

The BEA WebLogic Server allows organizations to extend the reach of their enterprise applications as Web services. BEA WebLogic provides J2EE specification, handling a wide range of common programming tasks for developers, including the provision of transaction services, guaranteed messaging, naming and directory services, database access and connection pooling, thread pooling, load balancing and fault tolerance.

For information on the J2EE standard, see the white paper titled “Obliv NetPoint: Centralized Security Management for BEA WebLogic J2EE Server Environments” available at <http://www.oblix.com/docs>.

The NetPoint Ready Realm for BEA integration with the BEA WebLogic Server has the following characteristics:

- Provides authentication to all BEA WebLogic resources (EJBs, servlets, and so forth)
- Provides fine-grained authorization to all BEA WebLogic resources (EJBs, Servlets, and so forth)
- Protects all resources on a BEA WebLogic Server (WL Servers, Command Line Admin Tools, WL Events, WL servlets, WL JDBC connection pools, WL passwords, WL JMS destinations, and WL JNDI contexts)
- Enforces J2EE container-based security

- Maps and manages the BEA WebLogic Server security constraints in NetPoint policies, including BEA WebLogic ACLs, groups, roles and users
- Provides single sign-on between WebGates and the BEA WebLogic Server

## **BEA WebLogic Portal**

A portal provides a single point of access to enterprise data and applications, presenting a unified and personalized view of that information to employees, customers, and business partners.

The BEA WebLogic Portal provides user management, group portals, delegated administration, portal and portlet management, personalization, and campaigns.

NetPoint Ready Realm for BEA integration with the BEA WebLogic Portal has the following characteristics:

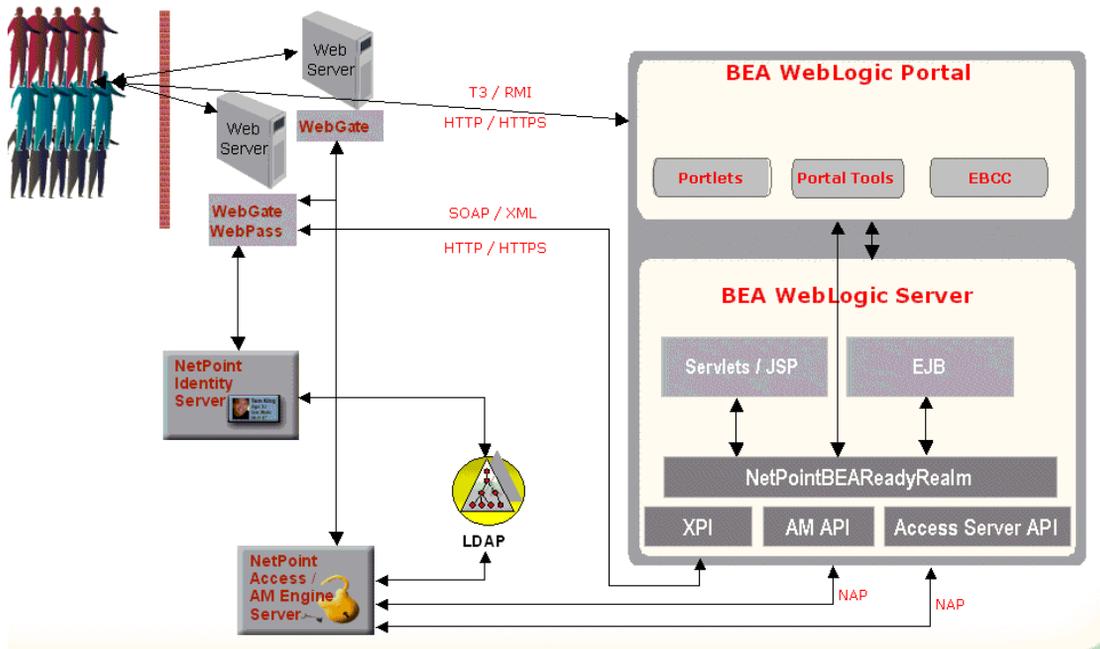
- Protects BEA WebLogic Portal resources with policies defined in the NetPoint Access Manager
- Enables creating and deleting users and groups from the BEA WebLogic Portal Admin tools
- Provides single sign-on between WebGates and portals

## **Integration Architecture**

The NetPoint Ready Realm for BEA is a NetPoint AccessGate that serves as a WebLogic Custom Security Realm. When installed and configured, it establishes a native connection between a BEA WebLogic Server and NetPoint.

Figure 1 shows a configuration that includes NetPoint Ready Realm for BEA.

**Figure 1** NetPoint Ready Realm for BEA Configured with BEA WebLogic Portal



See *NetPoint 7.0 Administration Guide Volume 2* for more information on AccessGate and Access Server configuration. For more information about the integration, see: “BEA WebLogic and NetPoint Objects” on page 31.

## BEA WebLogic and NetPoint Objects

NetPoint Ready Realm for BEA implements its own User, Group, and ACL classes to map BEA WebLogic Server objects to NetPoint objects. Table 1 maps common BEA WebLogic objects to their NetPoint equivalents.

**Note:** NetPoint policy domains, policies, default authentication and authorization rules, do not have an equivalent in WebLogic.

**Table 1** Mapping BEA WebLogic and NetPoint Objects

<b>WebLogic</b>	<b>NetPoint</b>
User	User Users are managed in the NetPoint COREid System
WebLogic System User	weblogic_system The NetPoint Ready Realm for BEA requires that the WebLogic System User be a workflow participant when creating and deleting users. This user is automatically created as weblogic_system during configuration of the NetPoint Ready Realm for BEA. You can change the weblogic_system user to another NetPoint user after initial installation of the NetPoint Ready Realm for BEA.
Guest User	The WebLogic Portal requires a guest user account. You can map this guest user account to any NetPoint user.
Group / Role	Implemented in the NetPoint Access Manager as a resource type of J2EE_Role. Access to (membership in) this role is determined through an authorization rule, which is linked to the J2EE role via a policy.  Resource(J2EE_Role) + Policy + Authz Rule  <b>Note:</b> The principal in BEA J2EE is either a user or a group. When WebLogic evaluates a security constraint, it looks for a user first and then looks for a group. If a user in NetPoint has the same name as a J2EE role, the user entity takes precedence when security constraints are enforced.  For example, if a user is named Administrator in NetPoint and a J2EE role is also named Administrator, the user Administrator will be used for all security constraint evaluations.
ACL	Implemented in the NetPoint Access Manager as a resource type of J2EE_ACL. Access to this resource type is determined through an authorization rule, which is linked to the J2EE_ACL via a policy.  Resource(J2EE_ACL) + Policy + Authz Rule
Permission	Resource Operation

When you first configure the NetPoint Ready Realm for BEA, the groups and ACLs required for the BEA WebLogic Portal and BEA WebLogic Server are automatically generated in the NetPoint Access Manager.

## Managing Users

After you have installed and configured the NetPoint Ready Realm for BEA, the directory server becomes the central repository for all user profiles. You can add users through the NetPoint COREid System or the BEA WebLogic Portal.

If desired, you can restrict this functionality in the BEA WebLogic Portal Admin tool.

## Managing Membership to BEA WebLogic Groups and Roles

After you have installed and configured the NetPoint Ready Realm for BEA, you can dynamically maintain group membership by defining default access rules and fine-grained authorization policies.

---

**Note:** BEA WebLogic groups and roles must be managed through the Access Manager (as opposed to the COREid Group Manager application). All J2EE\_Roles must reside in the same policy domain where they are defined by authorization rules within a policy. By default this policy domain is called *weblogic* and is configured during installation of the NetPoint Ready Realm for BEA. To configure a different policy domain for different WebLogic servers under SPPI, see “Configuring Multiple Policy Domains in NetPoint for Different WebLogic Servers” on page 168.

---

## Managing ACLs

The WebLogic ACLs must be managed through the Access Manager as a resource type. You can centrally define default access rules and fine-grained authorization policies to these ACLs.

## SSO to BEA WebLogic Resources

The NetPoint Ready Realm for BEA provides single sign-on between WebGates, the BEA WebLogic Portal (including applications within the portal framework) and BEA WebLogic Server Resources (EJBs, JSPs, Servlets).

To implement SSO with BEA WebLogic Server 6.x, and 7.x (in backward compatibility mode) in a NetPoint environment, the BEA WebLogic Server needs to be configured to use the NetPoint BEA Realm. The Login page (either .jsp file or a servlet) must be modified as illustrated in the chapter on form-based authentication in the *NetPoint 7.0 Administration Guide Volume 2*. See also “Single Sign-On (SSO) Servlet” on page 84.

# Supported Versions and Platforms

NetPoint supports the following versions of BEA WebLogic Application server and BEA WebLogic Portal server:

**Table 2** Supported WebLogic Versions

WebLogic Server	WebLogic Portal
Version 8.1 SP2	Version 7.0 (sp1)
Version 7.0 SP2	Version 4.0 (sp2)
Version 6.1 SP2	

NetPoint supports the BEA WebLogic Application server and WebLogic Portal server via SSPI on the following platforms:

- HP-UX 11.0i
- Red Hat Enterprise Linux AS 2.1
- Solaris 8
- Windows 2000 Advanced Server SP4

---

**Note:** Support for BEA WebLogic v6.1 is deprecated and is *not* planned for future releases of NetPoint.

---

## Configuring NetPoint Ready Realm for BEA

Configuring NetPoint Ready Realm for BEA includes several procedures.

### Task overview: Configuring Ready Realm

1. Prepare for integration, as described in “Preparing for Integration with NetPoint Ready Realm” on page 35.
2. Add an AccessGate for the NetPoint Ready Realm for BEA, as described in “Adding an AccessGate” on page 35.
3. Generate policies, workflows and properties automatically from the Access System Console, as described “Generating NetPoint Ready Realm for BEA Information” on page 37.
  - Set the password of the weblogic\_system user.
  - Enable the WebLogic Policy Domain.
  - Restart the Access Servers.

## Preparing for Integration with NetPoint Ready Realm

Before you can configure NetPoint Ready Realm for BEA, several components must be installed and confirmed to be running properly.

### To prepare for integration with NetPoint Ready Realm for BEA

1. Install the appropriate BEA WebLogic Server, as described in “Supported Versions and Platforms” on page 34 and your vendor documentation.  
<http://edocs.bea.com>
2. Install BEA WebLogic Portal Server, as described in “Supported Versions and Platforms” on page 34 and your vendor documentation.

---

**Note:** BEA WebLogic Server 6.1 SP1 is deprecated and may impede file synchronization for EBCC. See “Supported Versions and Platforms” on page 34.

---

3. NetPoint COREid System and NetPoint Access System, as described in the *NetPoint 7.0 Installation Guide*.

## Adding an AccessGate

You add an AccessGate for NetPoint Ready Realm for BEA through the NetPoint Access System Console.

### To add an AccessGate

1. From the Access System Console, click Access System Configuration.
2. Click AccessGate Configuration.
3. Click Add.
4. Enter a Name for this AccessGate.

Because this AccessGate is associated with the NetPoint Ready Realm for BEA, you may want to name it accordingly; for example, BEARealm

5. Enter the hostname.

---

**Note:** You do not need to enter a port for an AccessGate configured for NetPoint Ready Realm for BEA.

---

6. Set the primary cookie domain.  
This enables SSO for users accessing the WebLogic Server and Portal.
7. Set the Session Timeout.
8. Enter an AccessGate Password. (Optional)

9. Turn on Access Management Service.

This enables this AccessGate to use the Access Manager API, which is required by the NetPoint Ready Realm for BEA.

10. Select the appropriate Transport Security Mode, if it is different than Open.

See the chapter on AccessGate and Access Server configuration in *NetPoint 7.0 Administration Guide Volume 2* for more information on Transport Security Modes.

11. Save this AccessGate.

12. After an AccessGate is added, you will be prompted to add an Access Server.

The following example shows a completed AccessGate screen.

Logged in user: Lou Re

Oblix • NetPoint

System Configuration | NetPoint System Management | Access System Configuration

### Add a new NetPoint AccessGate

AccessGate Name	BEARrealm
Hostname	instructor2.oblix.com
Port	
Access Gate Password	*****
Re-type Access Gate Password	*****
Debug	<input checked="" type="radio"/> Off <input type="radio"/> On
Access Management Service	<input type="radio"/> Off <input checked="" type="radio"/> On
Maximum user session time (seconds)	1200
Idle Session Time (seconds)	1200
Primary HTTP Cookie Domain	.oblix.com
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

13. Select the appropriate Access Server from the drop-down menu.

**Note:** In order for the AccessGate to function, you must turn on the Access Management Service for the AccessGate, as well as all Access Servers associated with the AccessGate.

14. Install the AccessGate and Access Server, as described in the *NetPoint 7.0 Installation Guide*.

## Generating NetPoint Ready Realm for BEA Information

The NetPoint Ready Realm for BEA configuration functionality in the NetPoint Access System Console provides an automated way to create Resource Type definitions, policies, and workflows to support the BEA WebLogic Application Server and BEA WebLogic Portal in the NetPoint environment.

Auto-configuration consists of the following four screens:

- Screen 1:4
- Screen 2:4
- Screen 3:4
- Screen 4:4

### Screen 1:4

This step creates Resource Type definitions (such as J2EE\_Roles, J2EE\_ACLS) and a policy domain with Access System policies to support the BEA WebLogic Server and Portal.

During this step, the configuration tool also adds the `weblogic_system` user to the directory server. The `weblogic_system` user is created under the searchbase specified for the NetPoint Access System. This user is granted administrative rights for policies. In the next step, the user will be a participant in BEA Ready Realm workflows. By default, this user is mapped to the system user of the WebLogic Server in the properties files generated in the last step. It is possible to map an other user to the `weblogic_system` user after the NetPoint policies and workflows are generated. See “Advanced Configuration Options” on page 73 for details.

An authentication scheme and a resource of type `Authen` is added for authentication requests from the BEA WebLogic Server or WebLogic Portal.

The following J2EE\_Roles are added automatically by the configuration tool:

- SystemAdministrator
- AdminEligible
- DelegatedAdministrator
- Administrator
- Everyone

Policies are added to give only the `weblogic_system` user access to all the roles other than *everyone*. A policy is added to include all users in the *everyone* role.

---

**Note:** The `J2EE_Roles` in NetPoint map to both BEA WebLogic roles and BEA WebLogic groups.

---

`J2EE_ACLS` are added to protect BEA WebLogic Server resources. Policies are generated for the following ACLs:

- `weblogic.admin`
- `weblogic.event`
- `weblogic.jdbc.connectionPool`
- `weblogic.jms.queue`
- `weblogic.jms.topic`
- `weblogic.jndi`
- `weblogic.jndi.BEA_commerce`
- `weblogic.jndi.javax`
- `weblogic.jndi.p13nApp`
- `weblogic.jndi.portal`
- `weblogic.jndi.weblogic`
- `weblogic.jndi.wlcsApp`
- `weblogic.passwordpolicy`
- `weblogic.server`
- `weblogic.servlet`
- `weblogic.workspace`

A default authorization rule is added to grant access to all these policies only for the `weblogic_system` user. The `weblogic_system` user is made the Delegated Administrator of the policy domain. This policy domain is disabled by default; you must enable it before using the NetPoint Ready Realm for BEA in WebLogic servers.

---

**Note:** The access policies or workflows will not be re-generated or modified if they have already been generated. If you want to regenerate the policies, you must first delete the entire WebLogic policy domain and then regenerate the policies.

---

## Screen 2:4

In this step, you select a user as the administrator for NetPoint policies. This user is referred to as the `weblogic_system` user.

## Screen 3:4

This step automatically creates NetPoint workflows that will support creating and deleting users from within the WebLogic Portal Administrative interface.

For more information, see “Advanced Configuration Options” on page 73.

## Screen 4:4

This step automatically generates parameters that you will need to copy from the Access System Console Web page and add into a `NetPointBEARealm.properties` file. The `NetPointBEARealm.properties` file is generated during the installation of a NetPoint Ready Realm for BEA.

---

**Note:** You can access these parameters for the properties file at any time by clicking the NetPoint Ready Realm for BEA configuration button again. Accessing the parameters will not overwrite your modifications.

---

### To generate NetPoint Ready Realm for BEA

1. From the Access System Console, click Access System Configuration.
2. Click the NetPoint Ready Realm for BEA Configuration link on the side navigation bar.

The first of four configuration screens is displayed.

Obliv • NetPoint **System Configuration** NetPoint System Management Access System Configuration

**NetPoint BEA Ready realm Configuration**

The NetPoint BEA Ready Realm Configuration provides an automated way to create required files, workflows, resource type definitions and policies to support the BEA WebLogic Application Server and BEA WebLogic Portal in the NetPoint environment.

This configuration should only be completed by administrators in those environments who plan on installing the NetPoint BEA Ready Realm.

There are three steps to complete:

1. Creation of the Resource Type Definitions (J2EE\_Roles and J2EE\_ACLs) and the Access Manager policies to support the WebLogic Server and Portal.
 

In this stage we will also add a user called 'weblogic\_system' to the directory server. This user will be granted administrative rights for Policies and BEA Ready Realm specific Workflows. This user is mapped by default to the system user of the Weblogic Server.

It will be possible to change this user and select any other user to map to the Weblogic system user after the NetPoint Policies and Workflows are generated.
2. Creation of NetPoint Workflows that will support creating and deleting users from within the WebLogic Portal Administrative interface.
3. Generation of entries that you will need to copy from the web page and add into a NetPointBEARealm.properties file. The file will be generated during install of the NetPoint BEA Ready Realm.
 

The information generated during this stage should be saved in a text file so that you can add it after installation of the NetPoint BEA Ready Realm has been completed.

AccessGate Configuration  
 Access Server Configuration  
 Authentication Management  
 Authorization Management  
 User Access Configuration  
 Common Information Configuration  
 Host Identifiers  
 NetPoint BEA Ready Realm Configuration  
 Help  
 About  
 Logout

---

**Note:** If you are running Active Directory, you should wait 5 to 15 minutes before clicking Next in order to allow Active Directory’s caching update to take effect. See “Implementation Notes for Active Directory” on page 88 for more information.

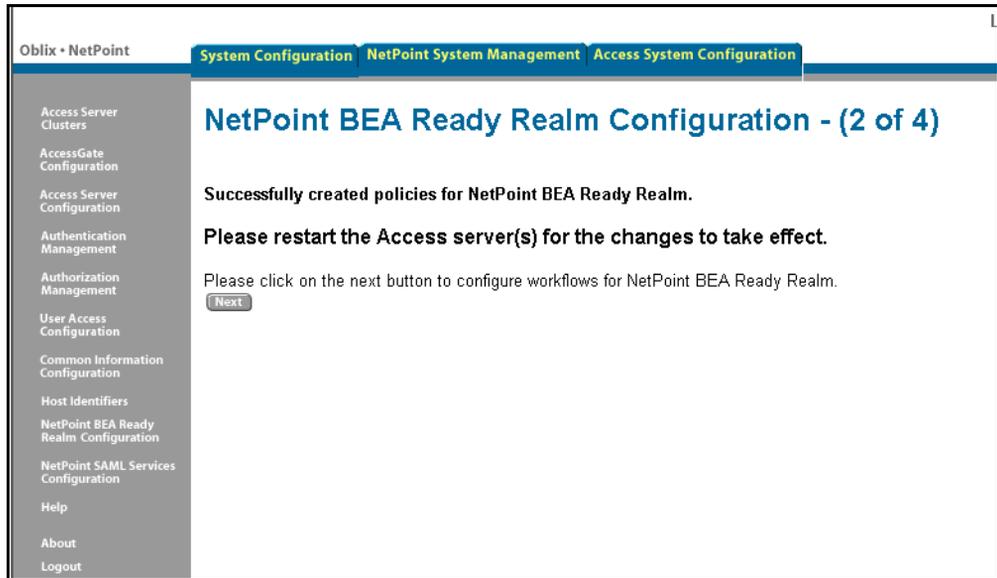
---

3. Click Next at the bottom of the screen.

You are prompted to select a user as the weblogic\_system user. See the *NetPoint 7.0 Administration Guide Volume 1* for information on using the Selector.

4. Click Next to generate NetPoint Ready Realm for BEA policies and resource types.

The following page appears.



You next create the `weblogic_system` user password.

### To create the `weblogic_system` user password

1. Launch a new browser window.
2. Access the NetPoint COREid User Manager application.
3. In the Search field, verify That Contains is selected as the search operator and enter `Web` as the search criteria.
4. Click Go.

The `weblogic_system` user is displayed.

5. Click the link for the `WebLogic_system` user to bring up the associated profile.
6. Click Modify.

You must have view and modify permissions to change an attribute value on a profile.

7. Change the password.

This does not have to be the same as the WebLogic Server system account password.

---

**Note:** By default there is no password created for the `weblogic_system` user. WebLogic Server requires a password in order to boot as this user.

---

You next set the serachbase and define attribute access controls for the `weblogic_system` user.

## To set the searchbase and define attribute access control for the weblogic\_system user

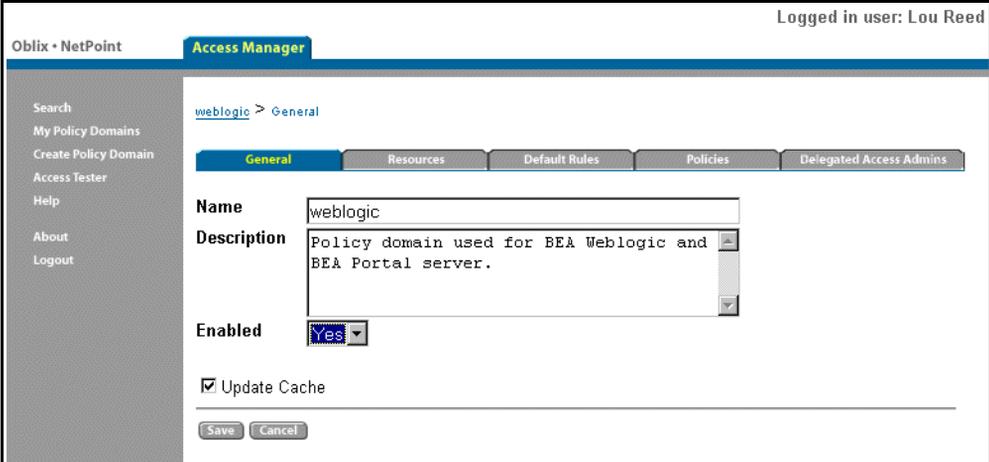
1. Define a searchbase for the weblogic\_system user. This searchbase should be the same as the default searchbase for the COREid System.
2. Define attribute access control to allow the weblogic\_system user Read/Write access to the attributes assigned to the Full Name and Login semantic types; for example, cn and uid.

See *NetPoint 7.0 Administration Guide Volume 1* for more information on configuring User, Group, and Organization Managers.

You next enable the WebLogic policy domain.

## To enable the WebLogic policy domain

1. From the NetPoint Access Manager, select the generated WebLogic policy domain, and click Modify.
2. Select Yes in the Enabled drop-down list.



The screenshot shows the NetPoint Access Manager interface. The top right corner indicates the user is logged in as 'Lou Reed'. The main navigation bar includes 'Obliv - NetPoint' and 'Access Manager'. A sidebar on the left contains links for Search, My Policy Domains, Create Policy Domain, Access Tester, Help, About, and Logout. The main content area shows the configuration for the 'weblogic' policy domain under the 'General' tab. The 'Name' field is set to 'weblogic'. The 'Description' field contains the text: 'Policy domain used for BEA Weblogic and BEA Portal server.'. The 'Enabled' dropdown menu is set to 'Yes'. There is a checked checkbox for 'Update Cache'. At the bottom, there are 'Save' and 'Cancel' buttons.

3. Click Save
4. From the browser where you were configuring NetPoint Ready Realm for BEA, click Next to automatically create workflows that will support creating and deleting users from the WebLogic Portal Administration tool.

The following confirmation screen appears.

Oblix • NetPoint Logged in user: M

**System Configuration** | **NetPoint System Management** | **Access System Configuration**

## NetPoint BEA Realm Configuration - (3 of 4)

### Successfully generated workflows for NetPoint BEA Realm

**The following workflow(s) have been generated:**

- Create type workflow - obworkflowid=wfq520030905T15574476417,obcontainerId=workflowDefinitions,ou=Oblix,ou=company,dc=clutch,dc=oblix,dc=com
- Delete type workflow - obworkflowid=wfq520030905T15574587621,obcontainerId=workflowDefinitions,ou=Oblix,ou=company,dc=clutch,dc=oblix,dc=com

Click next button to continue configuring NetPoint BEA Realm

- Access Server Clusters
- AccessGate Configuration
- Access Server Configuration
- Authentication Management
- Authorization Management
- User Access Configuration
- Common Information Configuration
- Host Identifiers
- NetPoint BEA Ready Realm Configuration
- NetPoint SAML Services Configuration
- Help
- About
- Logout

5. Click Next.

NetPoint generates entries that you need to copy into the `NetPointBEARealm.properties` file.

Logged in user: Ma

Oblix • NetPoint System Configuration NetPoint System Management Access System Configuration

---

Access Server Clusters

AccessGate Configuration

Access Server Configuration

Authentication Management

Authorization Management

User Access Configuration

Common Information Configuration

Host Identifiers

NetPoint BEA Ready Realm Configuration

NetPoint SAML Services Configuration

Help

About

Logout

## NetPoint BEA Ready Realm Configuration - (4 of 4)

**Configuration of NetPoint BEA Ready Realm is done. Please copy & append the following text to your NetPointBEARealm.properties file**

---

```

# This file contains partial configuration of NetPoint BEA Ready Realm
# Please copy the parameters below to NetPointBEARealm.properties file.

# The NetPoint BEA Ready Realm installer will create the NetPointBEARealm.properties file
# in $REALM_INSTALL_DIR/examples directory.

# Note: NetPoint BEA Ready Realm requires this file to be in the Weblogic home directory.
# For Weblogic 5.1:
# If weblogic was installed in C:\weblogic this file should be copied to c:\weblogic
# For Weblogic 6+:
# If weblogic was installed in C:\bea this file should be copied to c:\bea\wlserver6.0
# For BEA Portal server 4.x:
# If BEA Portal server was installed in C:\bea this file should be copied to c:\bea\wlportal4.0

# NetPoint WebPass webserver host name and port number
OBWebPass.hostname=yttrium.oblix.com
OBWebPass.port=80

# ID of the workflow to create users and the domain specified in this workflow
OBWebPass.CreateUserWorkFlowDomain=OU=Company,DC=qalab-clutch,DC=oblix,DC=com
OBWebPass.CreateUserWorkFlowID=obworkflowid=wfgs20030905T15574476417_obcontainerId=workflowDefinitions.ou=Oblix,ou

```

---

**Note:** You can access these parameters at any time from the Access System Console, by clicking NetPoint Ready Realm for BEA Configuration, and clicking Next to get to the properties screen. You will need to copy these parameters for each WebLogic Server.

---

## Installing NetPoint Ready Realm for BEA

NetPoint Ready Realm for BEA uses WebPass to make IdentityXML calls to create and delete users from the WebLogic Portal Administration tool using NetPoint workflows.

By default, the NetPoint Ready Realm for BEA will use the first WebPass listed. Typically, this will be on the same Web server that hosts the NetPoint Access Manager. If you have more than one WebPass and want NetPoint Ready Realm for BEA installed for a different WebPass, you can change the host machine and port in the NetPoint Ready Realm for BEA properties file after installation.

See topics below for additional information:

- “Preparing for Ready Realm Installation” on page 45
- “Installing NetPoint Ready Realm for BEA” on page 46

## Preparing for Ready Realm Installation

When running the NetPoint Ready Realm for BEA installation program, you must provide information about the various NetPoint components it communicates with, including the WebPass and Access Server.

### To prepare for Ready Realm installation

1. Determine if the WebPass that NetPoint Ready Realm for BEA will use is protected by a WebGate.

**Yes:** During NetPoint Ready Realm for BEA installation, you need to enter the cookie domain for that WebGate (for example, *mycompany.com*) and cookie path (for example, *\*). Entering this information during installation enables SSO between IdentityXML and WebGate.

**No:** If WebPass is not protected by WebGate during NetPoint Ready Realm for BEA installation, but you protect it at a later time, you must navigate to *install\_dir\NetPointBEARealm\Examples\NetPointBEARealm.properties* file and change the values to:

```
OBWebPass.isProtected=true
OBWebPass.cookieDomain=your domain
OBWebPass.cookiePath=yourCookiePath
```

2. Determine if the machine hosting WebPass is running SSL.

If so, this requires additional certificate setup between the browser and the Web server hosting WebPass. Specifically, you must set up the HTTP Client used by NetPointBEARealm to make IdentityXML requests that accept server certification and make HTTPS requests to WebPass.

3. Set up the HTTP Client, as discussed below:

- a) Add *jsse.jar*, *jnet.jar*, *jcrt.jar* to the CLASSPATH in the start scripts of BEA WebLogic and BEA Portal server.

See <http://www.weblogic.com/docs51/admindocs/classpath.html> for more information.

- b) Add *NetPointBEARealm\_install\_dir\NetPointBEARealm\oblix\lib* path to the CLASSPATH.
- c) Extract JSSE.zip SSL Patch for HTTPClient (from <http://www.innovation.ch/java/HTTPClient/JSSE.zip>) in *NetPointBEARealm\_install\_dir\NetPointBEARealm\oblix\lib*.
- d) Obtain certificate of WebServer and certificate of CA who has issued certificate for WebServer running SSL and place that in a file (say *server.cer* and *ca.cer*).

- e) Use keytool in %JAVA\_HOME%\bin or %JAVA\_HOME%\jre\bin to add ca.cer and server.cer certificates to jssecacert keystore:

```
keytool -import -alias ca -file ca.cer -keystore jssecacert
```

```
keytool -import -alias server -file server.cer -keystore jssecacert
```

- f) Copy this file to %JAVA\_HOME%\lib\security or %JAVA\_HOME%\jre\lib\security.

Specify whether you want caching for NetPoint user and J2EE role information for the NetPoint Ready Realm for BEA.

Specify the Transport Security Mode between the AccessGate and Access Server.

Enter identification information for the AccessGate and Access Server.

## Installing NetPoint Ready Realm for BEA

The installation procedure differs depending on the platform on which you are installing NetPoint.

### To launch the installation

1. Locate and launch the installation package for NetPoint Ready Realm for BEA.

For example:

```
./NetPoint7_0_sparc-s2_BEAREalm
```

After you have launched the installation, the rest of the procedure is common to all platforms.

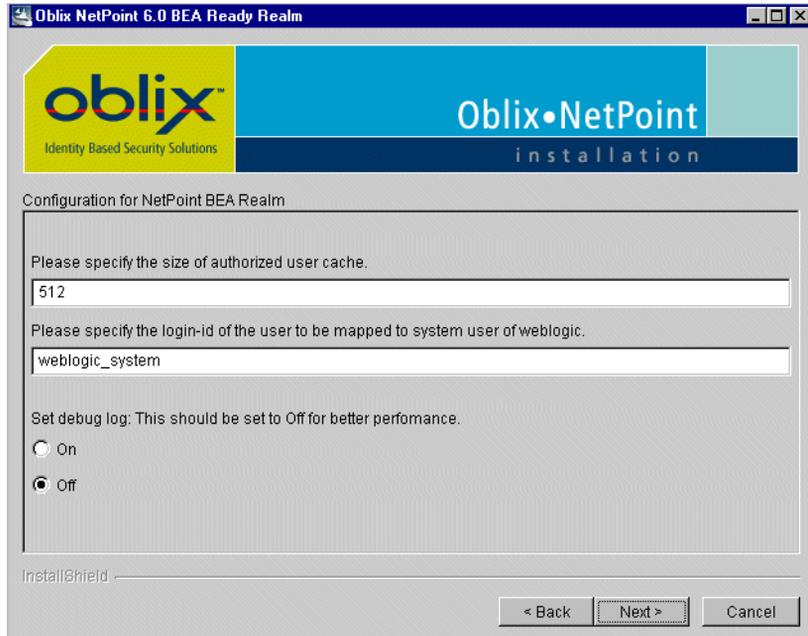
2. Specify the installation directory for the NetPoint Ready Realm for BEA.

The default is the directory containing your NetPoint installation.

3. Click Next.

4. Click Next again to copy the installation files.

The Configuration Information screen for NetPoint BEA Realm appears.



5. Specify the configuration information for the BEA Realm:

- Size of the user cache (default 512).

This should be set to the number of users expected during the peak load. This parameter controls the size of NetPoint Ready Realm for BEA's cache for authenticated users.

- Login ID of the NetPoint user to be mapped to the weblogic\_system user.

The default is weblogic\_system, which was created when NetPoint Ready Realm for BEA was configured. It is recommended that you complete installation and verify success before changing this user account. The password *does not* have to be the same as the weblogic\_system user's password.

- Set Debug Log.

When debug is set to on, messages will be reported to the console as well as the associated WebLogic Server's weblogic.log file. If you are working on the WebLogic default server *mydomain* you will find the log file under

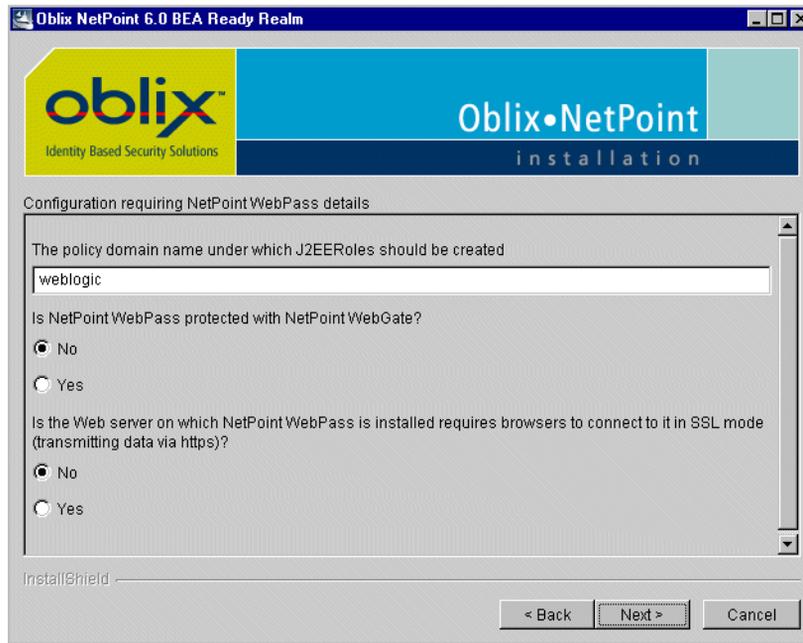
*BEA\_install\_dir*\wlserver\config\mydomain\logs

where *BEA\_install\_dir* is the directory where BEA WebLogic is installed. If you are working on the WebLogic Portal you will find the log file under

*BEA\_install\_dir*\wlportal\config\portalDomain\logs

6. Click Next.

The Configuration requiring NetPoint WebPass details screen appears.



7. Enter the following information:

- Policy domain under which J2EE Roles will be created.

When a WebLogic Portal administrator adds Groups using the WebLogic Portal product, the J2EE Roles will be created under the policy domain specified here. The default is weblogic (the policy domain generated using the NetPoint Ready Realm for BEA Configuration process in the Access System Console).

- Is NetPoint WebPass Protected with NetPoint WebGate?

Selecting Yes, will prompt you for the cookie domain and cookie path when you click Next.

Enter the cookie domain for that WebGate (for example, *.mycompany.com*) and cookie path (for example, */*). Entering this information during installation will enable SSO between IdentityXML and WebGate.

---

**Note:** If you have chosen to use WebGate to protect WebPass the assumption is that you are protecting the NetPoint applications with policy domains. It is also therefore, assumed that SSO between these components has been configured correctly.

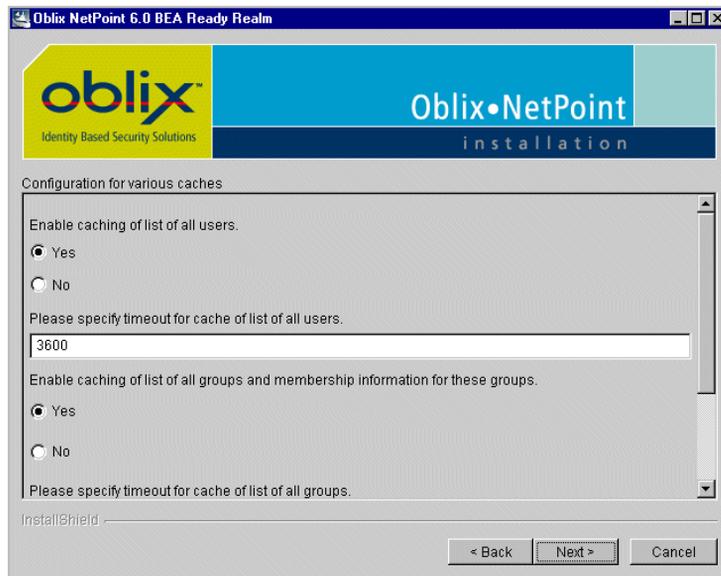
---

- Does WebPass require an SSL connection?

If so, this requires additional certificate setup between the browser and the Web server hosting WebPass. See the *NetPoint BEA Ready Realm Readme* file for more information.

8. Click Next.

This screen for configuring various caches appears.



9. Specify whether you want a caching list of all users.

This caches all NetPoint users for the NetPoint Ready Realm for BEA.

If you selected Yes, specify the timeout for the cache of listed users. The default is 3600 seconds. This means that it will be one hour before the cache is refreshed.

---

**Note:** When changes are made to user information in WebLogic, such as adding users or deactivating users, the changes are displayed immediately in the NetPoint cache. However, changes made to user information in NetPoint are not reflected in the WebLogic cache until the cache is refreshed. To ensure that you are viewing the most recent user-related information from NetPoint, refresh the WebLogic cache.

---

10. Specify whether you want a caching list of all groups and membership information for these groups.

This caches all NetPoint J2EE\_Role information and lists of all users who are members of these J2EE\_Roles for the NetPoint Ready Realm for BEA.

---

**Note:** NetPoint COREid System groups are not the same as WebLogic groups. See “Configuring NetPoint Ready Realm for BEA” on page 34 for more information.

---

If you selected Yes, specify the timeout for the cache of listed groups. The default is 3600 seconds. This means that it will be one hour before the cache is refreshed.

---

**Note:** Use caution when changing the cache timeouts, as this can impact system performance.

---

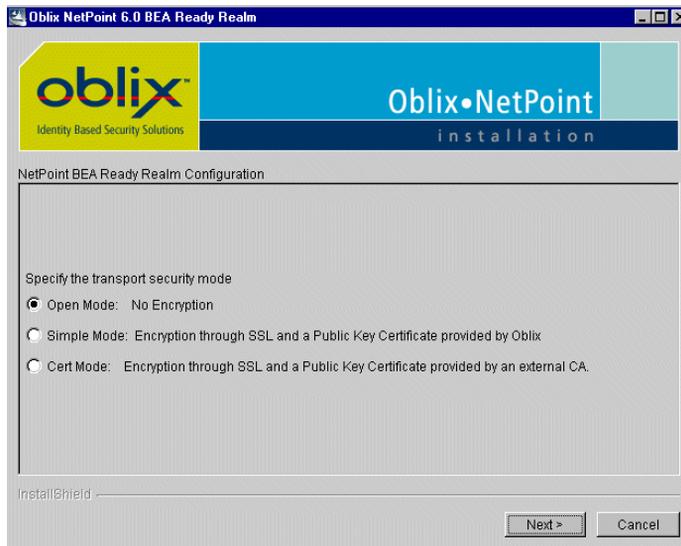
11. Specify the login ID of the user to be mapped to the guest user in WebLogic.

The default is guest. You can, however, map that to any valid NetPoint user using the value for his or her login ID.

The value you enter for your login ID will vary depending on the attribute you selected as the Login for NetPoint. See the chapter on COREid System Administration in *NetPoint 7.0 Administration Guide Volume 1* for more information on configuration object classes.

12. Click Next.

The NetPoint Ready Realm for BEA Configuration screen appears.



13. Enter the Transport Security Mode and click Next.

Specify the transport security mode between the Access Server and the NetPoint Ready Realm for BEA AccessGate. Clicking Next displays the AccessGate Configuration screen, as seen in the next figure.

---

**Note:** Subsequent screens vary depending on the selected Transport Security Mode. See the *NetPoint 7.0 Installation Guide* for information on each selection.

---

14. Enter the following information on the AccessGate Configuration screen:
  - **AccessGate ID**—Enter the name you specified earlier when adding an AccessGate in the Access System Console.
  - **Password for AccessGate**—Enter the AccessGate password you specified earlier when adding an AccessGate in the Access System Console, if applicable.
  - **Access Server ID**—Enter the name of the Access Server you associated with this AccessGate.

See the chapter on Access Server and AccessGate configuration in the *NetPoint 7.0 Administration Guide Volume 2* for more information.

You can specify any Access Server associated with the AccessGate entered above.
  - **Hostname where Access Server is installed**—Enter the hostname of the Access Server you associated with this AccessGate.

- **Port Number Access Server Listens To**—Enter the port of the Access Server you associated with this AccessGate.

15. Click Next to display the summary screen and then click Finish.

---

**Note:** If the Installation fails or you need to change these settings at a later time, you can run the `configureAccessGate` tool located in

`NetPointBEARealm_install_dir\NetPointBEARealm\oblix\tools\configureAccessGate`

where `NetPointBEARealm_install_dir` is the directory where NetPoint Ready Realm for BEA was installed.

---

## Configuring WebLogic for Ready Realm

Configuring WebLogic to communicate with the NetPoint Ready Realm for BEA involves the following procedures for your environment:

- “Appending the `NetPointBEARealm.properties` File” on page 53
- “Configuring NetPoint Ready Realm for WebLogic 6.x” on page 54
- “Configuring NetPoint Ready Realm for WebLogic 7.x” on page 59
- “Modifying WebLogic Server and Portal Startup Scripts” on page 63

## Appending the NetPointBEARealm.properties File

The NetPointBEAReadyRealm.properties file contains parameters that enable NetPoint Ready Realm for BEA to operate properly.

### To set NetPoint Ready Realm for BEA operation parameters

1. From the Access System Console, click Access System Configuration.
2. Click the NetPoint BEA Ready Realm Configuration link on the side navigation bar.

The Information page appears.

3. Click Next.

The NetPoint BEA Ready Realm Configuration page (1 of 4) appears.

4. Click Select User.

The Selector page appears.

5. Select a weblogic\_system user to delegate the administration rights for BEA policies.

See the *NetPoint 7,0 Administration Guide Volume 1* for information on using the Selector page.

6. Click Next to generate policies for NetPoint BEA Ready Realm.

On the next page (2 of 4), a message appears stating that policies have been created.

7. Click Next to generate workflows for NetPoint BEA Ready Realm.

On the next page (3 of 4), a message appears stating that workflows have been generated.

8. Click Next to continue with the configuration.

9. On the next page (4 of 4), select and copy all of the generated text, from beginning text to ending text.

Beginning text: # This file contains partial configuration of NetPoint Ready Realm for BEA.

Ending text: OBWebPass. DelUserWorkFlowComment=Deactivated user from WebLogic server.

10. From a text editor, open the NetPointBEARealm.properties file.

*NetPointBEARealm\_install\_dir*\examples\NetPointBEARealm.properties  
where *NetPointBEARealm\_install\_dir* is the directory where Ready Realm is installed. For example:

c:\NetPoint\NetPointBEARealm\examples.

You can access these properties at any time from the Access System Console, by clicking NetPoint BEA Ready Realm Configuration, and clicking Next to get to the properties screen.

11. Position the cursor at the end of the file and paste the property text in the file.

---

**Note:** For Windows, edit the OBAccessSDK.InstallDir parameter making sure there are double slashes in the path (for example, OBAccessSDK.InstallDir=D:\\NetPoint\\NetPointBEARealm). This is a known issue for Windows in this release.

---

12. Save the NetPointBEARealm.properties to the WebLogic Server and WebLogic Portal home directories. For example,

- C:\bea\wlserver
- C:\bea\wlportal

13. **Active Directory with Multiple Domains**—Set the OBWebPass.ADDomain property in the NetPointBEARealm.properties file, as indicated below:

- a) Specify the domain to which the system\_user belongs.
- b) Select the appropriate authentication scheme.
- c) Set the policy domain to WebLogic.
- d) Use the policy defined for /Authen/Basic resource.

See “Implementation Notes for Active Directory” on page 88 for details.

14. Restart the Access Server to allow the changes to take effect.

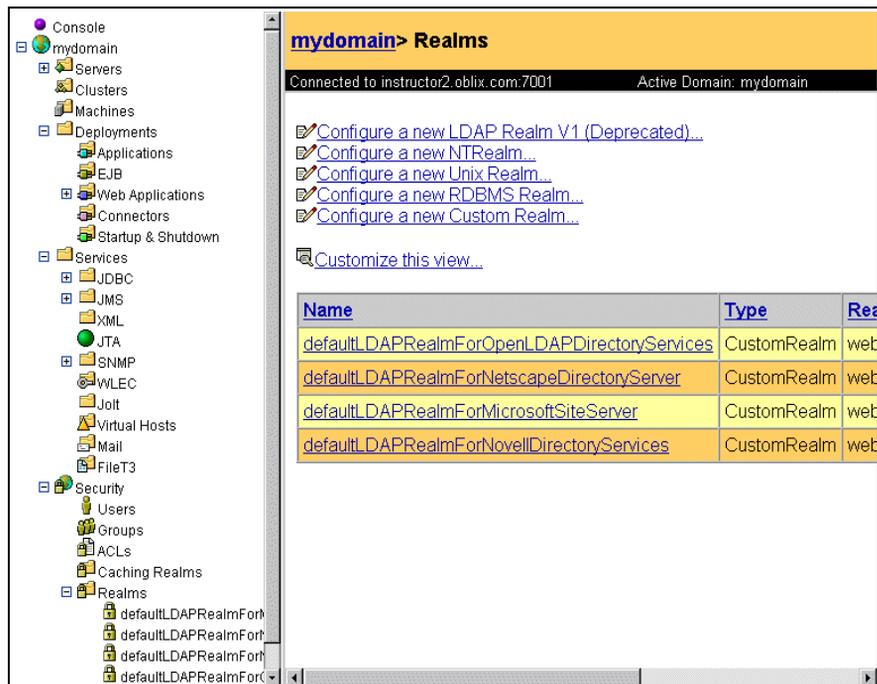
## Configuring NetPoint Ready Realm for WebLogic 6.x

You must enable NetPoint Ready Realm for BEA as a custom realm on each of your WebLogic servers. This requires that you know which WebLogic Server you will be using. You complete the following procedures:

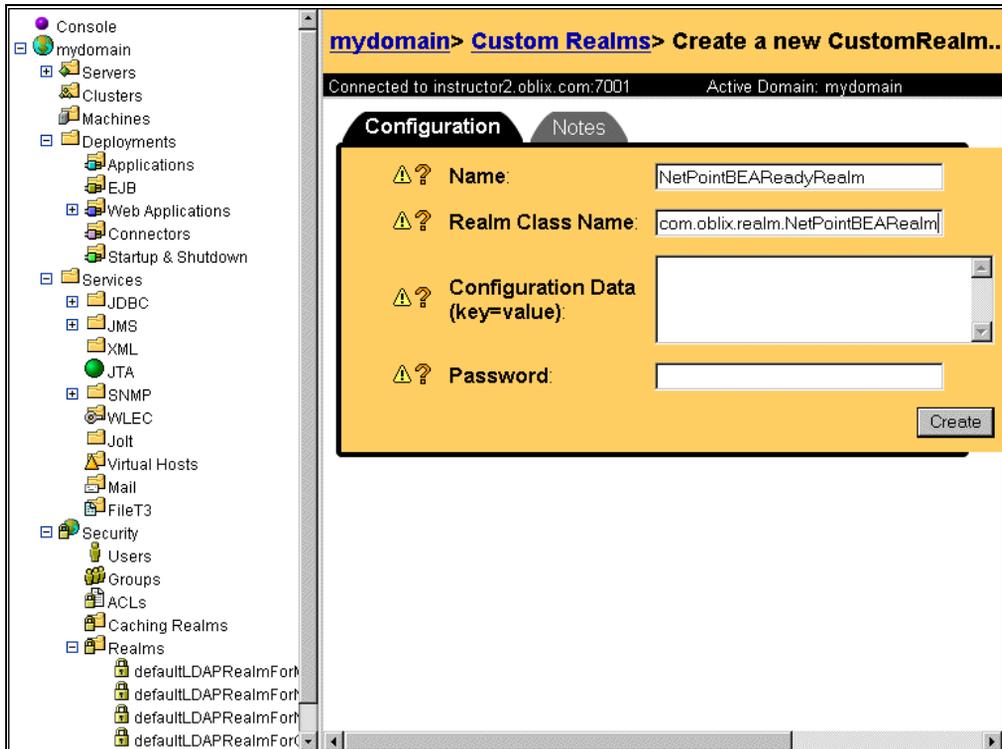
- “To create NetPoint Ready Realm for BEA as a custom realm” on page 55
- “To create a caching realm for the custom realm” on page 56
- “To create a security file realm for the caching realm” on page 58
- “To disable ACLs from the Realm.properties file” on page 58

## To create NetPoint Ready Realm for BEA as a custom realm

1. Start the WebLogic Server.
2. Launch the appropriate WebLogic Server console.
3. Login as the system user.
4. Expand the Security icon on the left navigation bar.



5. Click Configure a New Custom Realm on the right. For more information on configuring WebLogic Custom Realms, see:
  - WebLogic Administration Console.  
<http://edocs.bea.com/wls/docs61/adminguide/cnfgsec.html#1069864>
  - Configuring the Caching Realm  
<http://edocs.bea.com/wls/docs61/adminguide/cnfgsec.html#1063315>
6. WebLogic Administration Console. In the Name field, enter a name for the Custom Realm, for example, NetPointBEAReadyRealm.
7. In the Realm Class Name field, enter com.oblix.realm.NetPointBEARealm.

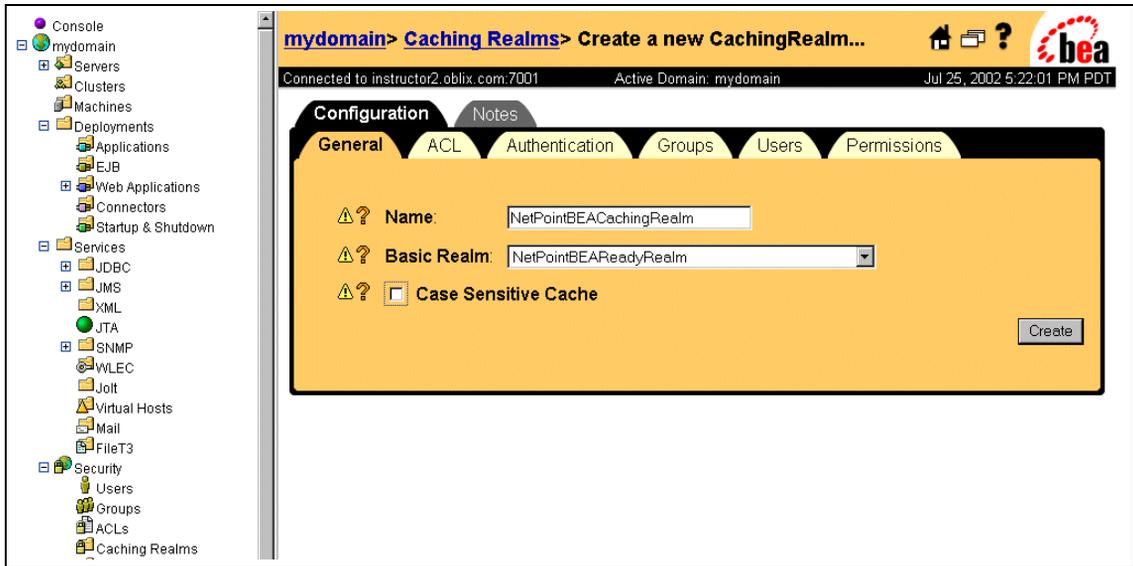


8. Click Create.
9. Click Apply.

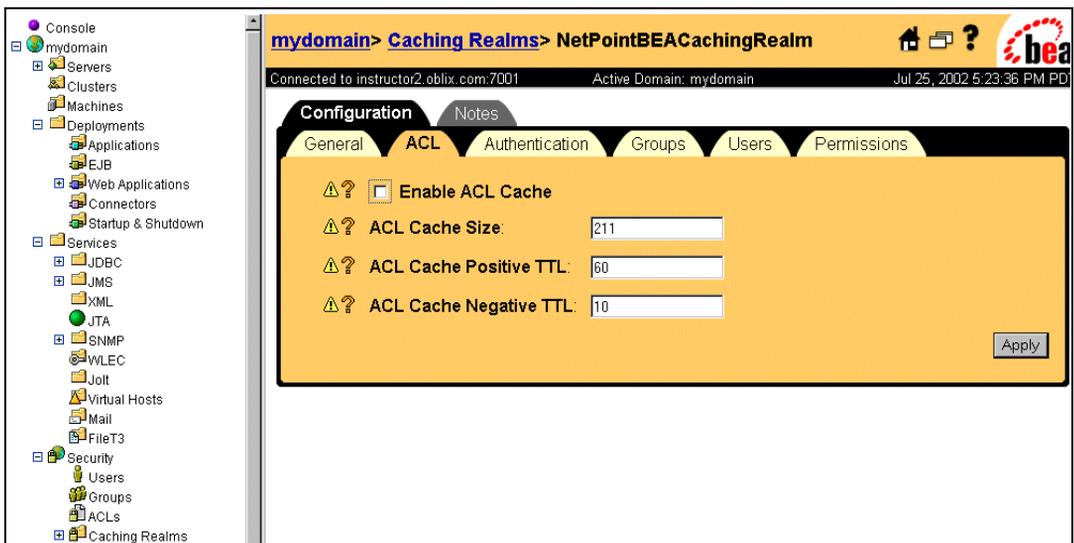
### To create a caching realm for the custom realm

1. Click on the Caching Realms icon in the left navigation bar.
2. Click Configure a New Caching Realm.
3. In the name field, enter the name of the realm you created; for example, NetPointBEACachingRealm.
4. Select NetPointBEARReadyRealm as the Basic Realm.
5. Select the appropriate case sensitive cache option for your NetPoint installation.

If your login ID in NetPoint is case sensitive, then the cache option also has to be case-sensitive.



6. Click Create.
7. Click Apply.
8. Click the ACL tab.
9. Deselect Enable ACL Cache.
10. Click Apply.



11. Repeat step 1 through step 10 for each tab.
12. Click on the Caching Realms icon in the left navigation bar.

- Verify that all caches are set to False.

Name	Basic Realm	Enable ACL Cache	Enable Authentication Cache	Enable Group Cache	Enable User Cache
NetPointBEACachingRealm	NetPointBEAReadyRealm	false	false	false	false

### To create a security file realm for the caching realm

- Click on the Security icon in the left navigation bar.
- Click the File Realm tab.
- Select the Caching Realm you created earlier; for example, NetPointBEACachingRealm

Configuration Security Notes

General File Realm Passwords Advanced

⚠️ Caching Realm: NetPointBEACachingRealm

⚠️ Max Users: 1000

⚠️ Max Groups: 1000

⚠️ Max ACLs: 1000

Apply

- Click Apply.
- Exit the WebLogic Console.

### To disable ACLs from the Realm.properties file

- Open the Realm.properties file with a text editor.
  - On the WebLogic default server, the Realm.properties file is in the c:\bea\wlserver6.1\config\mydomain directory.

- On the WebLogic Portal server, the Realm.properties file is in the c:\bea\wlserver4.0\config\portalDomain\directory.
2. Comment out the `acl.*` entries.  
You can use Search and Replace searching for `acl.` and replacing with `#acl.`
  3. Save your changes.

## Configuring NetPoint Ready Realm for WebLogic 7.x

Examples of configuring BEARealm and WebLogic 7.0 for compatibility are available in a sample file at the following location:

```
install_dir\examples\webapp\security_wl7.0
```

where *install\_dir* in the directory where you installed BEA Ready Realm.

The readme file explains the changes to the sample file that have been made for compatibility with WebLogic 7.0.

For instructions on how to set up NetPoint BEARealm to work with WebLogic 7.0, read the following sections:

- “Setting up NetPoint Ready Realm for BEA for Compatibility with WebLogic 7.0” on page 59
- “Setup for Compatibility with Weblogic Portal Server 7.0” on page 62

## Setting up NetPoint Ready Realm for BEA for Compatibility with WebLogic 7.0

You must configure NetPoint Ready Realm for BEA for compatibility with WebLogic 7.0.

### To set up NetPoint Ready Realm for BEA to run with WebLogic 7

1. Install WebLogic 7.0.
2. Using the WebLogic 7 Domain Configuration Wizard, create instances of a new WebLogic 7 domain; for example, *mydomain*, and a new WebLogic 7 server; for example, *myserver*.

---

**Note:** *Do not* start the new server before performing step 3.

---

3. Modify the config.xml file for the new server.

The config.xml file can be found in one of the following locations:

/bea/user\_projects/mydomain

or

c:\bea\user\_projects\mydomain

Be sure the config.xml file contains the following information before starting the server for the first time:

```
<Security Name="mydomain" Realm="mysecurity"/>
```

```
<Realm Name="mysecurity" FileRealm="myrealm"/>
```

```
<FileRealm Name="myrealm"/>
```

See also <http://edocs.bea.com/wls/docs70/upgrade/upgrade6xt070.html#1031626> for more information.

4. Start the WebLogic server using the start script, startWebLogic.cmd.
5. Log in to the system console as a user:

`http://hostname:port/console`

where *hostname* is the name of the host machine and *port* is the port number of the host machine.

6. Expand the Compatibility Security icon.
7. Add a new Realm called NetPointBEARealm with the class name com.oblix.realm.NetPointBEARealm
8. Add a new Caching Realm called NetPointBEACachingRealm.  
Be sure you specify the Basic Realm to be NetPointBEARealm.
9. Specify case sensitive cache and disable all ACL, Authentication, Group, User, and Permission caches.
10. Click the Compatibility Security icon, and do the following:
  - Disable guest
  - On the FileRealm tab, change CachingRealm to NetPointBEACachingRealm.
11. Locate the startWLS script, which is in the following location:

F:\bea\weblogic700\server\bin\startWLS.bat

or

/bea/weblogic700/server/bin/startWLS.sh

12. Add the following lines to the script:

**Windows:**

```
set NP_HOME=NetPointBEARealm_install_dir
```

where *NetPointBEARealm\_install\_dir* is the directory where the product was installed, for example, F:\NetPoint7.0\NetPointBEARealm.

```
set EXT_POST_CLASSPATH=%NP_HOME%\oblix\lib\HTTPClient.jar;  
%NP_HOME%\oblix\lib\jcert.jar;
```

```
%NP_HOME%\oblix\lib\jnet.jar;%NP_HOME%\oblix\lib\jobaccess.jar;  
%NP_HOME%\oblix\lib\jsse.jar;%NP_HOME%\oblix\lib\NetPointBEARealm.jar;
```

```
set PATH=.;%WL_HOME%\server\bin;%JAVA_HOME%\bin;%PATH%;  
%NP_HOME%\oblix\lib
```

**Unix:**

set the NP\_HOME, EXT\_POST\_CLASSPATH and PATH as

```
NP_HOME=NetPointBEARealm_install_dir
```

where *NetPointBEARealm\_install\_dir* is the directory where the product was installed, for example, F:\NetPoint7.0\NetPointBEARealm.

```
set EXT_POST_CLASSPATH=$NP_HOME/oblix/lib/  
HTTPClient.jar:$NP_HOME/oblix/lib/jcert.jar:  
$NP_HOME/oblix/lib/jnet.jar:$NP_HOME/oblix/lib/jobaccess.jar:  
$NP_HOME/oblix/lib/jsse.jar:  
$NP_HOME/oblix/lib/NetPointBEARealm.jar;
```

```
set PATH=.:$WL_HOME/server/bin:$JAVA_HOME/bin:$PATH:$NP_HOME/  
oblix/lib:
```

13. Place the NetPointBEARealm.properties file in the same directory as the config.xml file:

```
/bea/user_projects/mydomain
```

or

```
c:\bea\user_projects\mydomain
```

14. Restart the WebLogic server.

## Setup for Compatibility with Weblogic Portal Server 7.0

The following procedure describes setting up NetPoint Ready Realm for BEA for compatibility with WebLogic Portal Server 7.0.

### To set up Ready Realm to run with WLP 7.0

1. Install WebLogic Portal 7.0.
2. Using the WebLogic Server 7.0 Domain Configuration Wizard, create instances of a new WebLogic Portal 7.0 domain; for example, portalDomain, and a new WebLogic 7.0 server; for example, portalServer.
3. Start the WebLogic Portal using the start script StartPortal.bat.
4. Log in to the system console as a user as shown below:

```
http://hostname:port/console
```

where, *hostname* is the name of the host machine and *port* is the port number of the host machine.

5. Expand the Compatibility Security icon.
6. Add a new Realm called NetPointBEARealm with the class name com.oblix.realm.NetPointBEARealm.
7. Add a new Caching Realm called NetPointBEACachingRealm.
8. Be sure you specify the Basic Realm to be NetPointBEARealm.
9. Specify case-sensitive cache and disable all ACL, Authentication, Group, User, and Permission caches.
10. Click the Compatibility Security icon, and do the following:

- On the FileRealm tab, change CachingRealm to NetPointBEACachingRealm.
- Locate the startWLS script, which is in the following location:

```
F:\bea\weblogic700\portal\bin\win32\startWeblogic.cmd
```

or

```
/bea/weblogic700/portal/bin/solaris/startWeblogic.sh
```

- Add the following lines to the script to set NP\_HOME=NetPointBEARealm\_install\_dir:

where *NetPointBEARealm\_install\_dir* is where the product was installed, for example: F:\NetPoint7.0\NetPointBEARealm.

```
set CLASSPATH=%NP_HOME%\oblix\lib\HTTPClient.jar;%NP_HOME%\oblix\lib\jcert.jar;
```

```
%NP_HOME%\oblix\lib\jnet.jar;%NP_HOME%\oblix\lib\jobaccess.jar;
```

```
%NP_HOME%\oblix\lib\jsse.jar;%NP_HOME%\oblix\lib\NetPointBEARealm.jar;  
%CLASSPATH%  
set PATH=.;%NP_HOME%\oblix\lib;%PATH%;
```

11. Place the NetPointBEARealm.properties file in the same directory as the config.xml file:

/bea/user\_projects/portalDomain

or

c:\bea\user\_projects\portalDomain

12. Add the following J2EE\_Role in Access Manager and make sure that the user mapped to the weblogic\_system user is defined in these roles.
  - Deployers
  - Monitors
  - Operators
  - Administrators
13. Make sure that J2EE\_Acl for weblogic.jdbc.connectionPool authorizes everyone.
14. Restart the Weblogic server.

## Modifying WebLogic Server and Portal Startup Scripts

To modify the WebLogic Server startup script, you must include the appropriate .jar files in the classpath for each WebLogic Server. You will also need to add them to the path as in the following procedures:

- “To modify the WebLogic Server startup script” on page 63
- “To modify the WebLogic Portal Server startup script” on page 64

### To modify the WebLogic Server startup script

1. Locate the start weblogic.cmd file or the .bat file created for the WebLogic Server.

For example, C:\bea\wlserver\config\mydomain\  
startweblogic.cmd.

2. Locate set classpath, insert your cursor at the end of the existing entry, and add the following .jar files to the classpath. The files can be found in the *NetPointBEARealm\_install\_dir\oblix\lib*:
  - HTTPClient.jar
  - jnet.jar

- jsse.jar
- jcert.jar
- jobaccess.jar
- NetPointBEARealm.jar

For example:

```
Set CLASSPATH=D:\NetPoint\NetPointBEARealm\oblix\lib
\HTTPClient.jar;D:\NetPoint\NetPointBEARealm\oblix\lib\jnet;D
:\NetPoint\NetPointBEARealm\oblix\lib\jsse.jar;D:\NetPoint\Ne
tPointBEARealm\oblix\lib\jcert.jar;D:\NetPoint\NetPointBEARea
lm\oblix\lib\jobaccess.jar;
D:\NetPoint\NetPointBEARealm\oblix\lib\NetPointBEARealm.jarCL
ASSPATH
```

- If you are running WebPass in SSL mode, extract JSSE.zip to a directory and add that directory to the CLASSPATH as well. For example, D:\NetPoint\NetPointBEARealm\oblix\lib.

3. Add the following directories to the PATH:

*NetPointBEAReadyRealm\_install\_dir*\oblix\lib

where *NetPointBEAReadyRealm\_install\_dir* is the directory where the product was installed, for example, D:\NetPoint\NetPointBEARealm\oblix\lib or /NetPoint/NetPointBEARealm/oblix/lib.

4. Save the changes.
5. Shut down the WebLogic Server and restart it.

### **To modify the WebLogic Portal Server startup script**

1. Locate the startPortal.bat file.

For example, C:\bea\wlportal4.0\config\portalDomain\startPortal.bat.

2. Add the following .jar files to the CLASSPATH.

The files are in *NetPointBEAReadyRealm\_install\_dir*\oblix\lib:

- HTTPClient.jar
- jnet.jar
- jsse.jar
- jcert.jar
- jobaccess.jar

- NetPointBEARealm.jar

For example:

**Windows:**

```
set CLASSPATH=D:\NetPoint\NetPointBEARealm\oblix\lib\
NetPointBEARealm.jar;D:\NetPoint\NetPointBEARealm\oblix\lib\HTT
PClient.jar;D:\NetPoint\NetPointBEARealm\oblix\lib\jnet.jar;D:\
NetPoint\NetPointBEARealm\oblix\lib\jsse.jar;D:\NetPoint\NetPoi
ntBEARealm\oblix\lib\jcert.jar;D:\NetPoint\NetPointBEARealm\obl
ix\lib\jobaccess.jar;D:\NetPoint\NetPointBEARealm\oblix\lib\Net
PointBEARealm.jar;%CLASSPATH%
```

**Unix:**

```
set CLASSPATH=/NetPoint/NetPointBEARealm/oblix/lib/
NetPointBEARealm.jar:/NetPoint/NetPointBEARealm/oblix/lib/
HTTPClient.jar:/NetPoint/NetPointBEARealm/oblix/lib/jnet.jar:/
NetPoint/NetPointBEARealm/oblix/lib/jsse.jar:/NetPoint/
NetPointBEARealm/oblix/lib/jcert.jar:/NetPoint/
NetPointBEARealm/oblix/lib/jobaccess.jar:/NetPoint/
NetPointBEARealm/oblix/lib/NetPointBEARealm.jar:$CLASSPATH
```

3. Save the changes.
4. Locate the set-environment.bat file (bea\wlportal4.0\bin\win32\set-environment.bat) and add the following directories to the PATH:

*NetPointBEAReadyRealm\_install\_dir\oblix\lib*

For example, SET PATH=%PATH%;%WEBLOGIC\_HOME%\bin

For example, D:\NetPoint\NetPointBEARealm\oblix\lib.

5. Save the changes and shut down the WebLogic Server.
6. Restart the WebLogic Server.

## Protecting Resources With Policy Domains

To protect WebLogic resources with NetPoint policies, you must protect WebLogic URLs with the Web Application Descriptor. You do this by configuring a security constraint that defines the URLs that you want to protect and the roles that are allowed to access them. In NetPoint you define J2EE\_Acls, J2EE\_Roles, and authorization rules to allow access to NetPoint users and group.

When a user attempts to access a protected resource (URL), NetPoint checks if the user is included in the role that is defined for the URL. If authentication is successful, NetPoint authorizes the user to access the resources.

See the *NetPoint 7.0 Administration Guide Volume 2* for information about protecting resources with a policy domain.

There are many ways you can protect an EJB or a WebApp in NetPoint. The following section provides examples for protecting an EJB or a WebApp with a J2EE Role or with a J2EE ACL.

## Example: Defining a J2EE\_Role to protect an EJB or a WebApp

You can protect an EJB or a WebApp by defining a J2EE\_Role. You can specify security constraint for application roles to access certain methods in the deployment descriptor for the EJB method-level constructor that can be found in the `ejb-jar.xml` or `web.xml` files.

These application roles can then be mapped to NetPoint J2EE\_Roles in `weblogic-ejb.jar.xml` (or `weblogic.xml`) file. The principal specified in this file can be a NetPoint J2EE\_Role or a NetPoint user. Only users specified in the authorization rule for that J2EE\_Role will have access to specified methods.

---

**Note:** If you prefer to use a programmatic check for a user or role, try using one of these functions: `isCallerInRole()` for `EJB.Context` or `isUserInRole()` for `HttpServletRequest`. The referenced role must be declared in the corresponding XML file.

---

### Example 1

Defining a J2EE\_Role with a `ejb-jar.xml` file

```
<ejb-jar>
<enterprise-beans>
    <session>
<ejb-name>statelessSession</ejb-name>
    </session>
</enterprise-beans>
    <assembly-descriptor>
    <security-role>
    <description>AppBroker</description>
    <role-name>AppBroker</role-name>
    </security-role>
    </assembly-descriptor>
</ejb-jar>
```

### Example 2

Defining a J2EE\_Role with a `weblogic-ejb-jar.xml` file (`weblogic.xml` for a webapp):

```
<weblogic-ejb-jar>
```

```

        <weblogic-enterprise-bean>
    <ejb-name>statelessSession</ejb-name>
    </weblogic-enterprise-bean>
<security-role-assignment>
    <role-name>
        AppBroker
    </role-name>
    <principal-name>
        Broker
    </principal-name>
</security-role-assignment>
</weblogic-ear-jar>

```

### Example 3

The following example illustrates how to define a J2EE\_Role with a web.xml file to protect /index.html. The Authorized Role is AppBroker, defined in the <Security\_role> section. This role is mapped to a J2EE\_Role in NetPoint called Broker.

```

<web-app>
    <security-constraint>
    <web-resource-collection>
    <web-resource-name>Protected page</web-resource-name>
    <description>

```

These pages are only accessible by authorized users

```

    </description>
    <url-pattern>/protected/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>PUT</http-method>
    <http-method>POST</http-method>

```

```

</web-resource-collection>

```

```

<auth-constraint>
    <description>

```

These are the roles that have access

```

    </description>
    <role-name>
        AppBroker
    </role-name>
</auth-constraint>

```

```

</security-constraint>

```

```

<security-role>
    <description>

```

```

        An AppBroker role
    </description>
    <role-name>
        AppBroker
    </role-name>
</security-role>
</web-app>

```

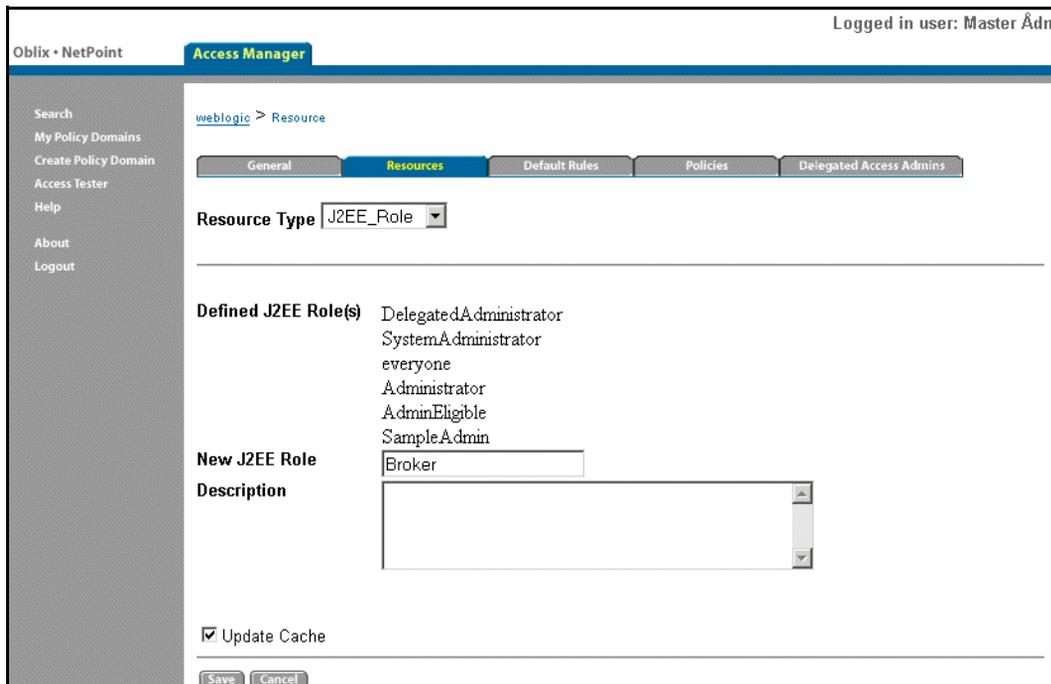
In the weblogic.xml file:

```

<weblogic-web-app>
    <security-role-assignment>
        <role-name>
            AppBroker
        </role-name>
        <principal-name>
            Broker
        </principal-name>
    </security-role-assignment>
</weblogic-web-app>

```

1. In NetPoint you must create a J2EE\_Role called Broker as shown below.



## 2. Add a policy for this J2EE\_Role.

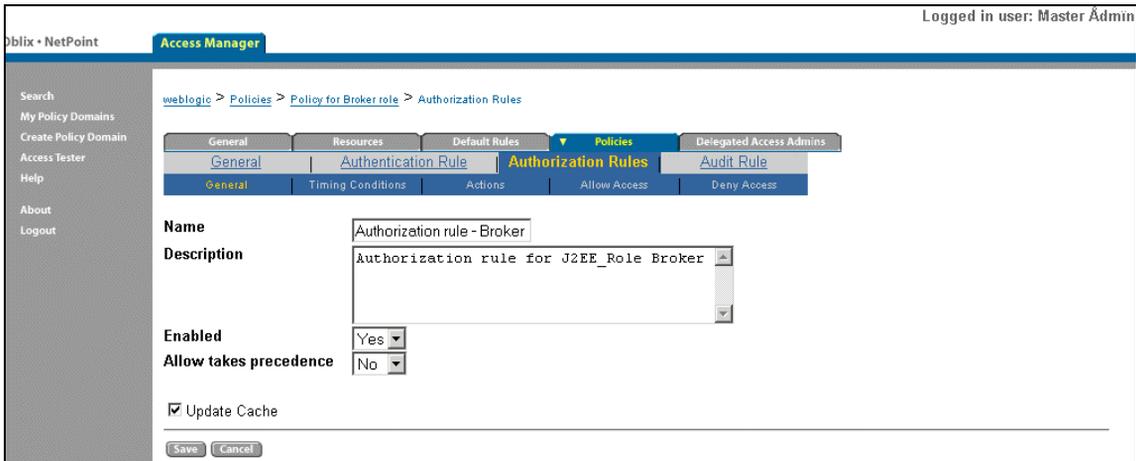
The screenshot shows the NetPoint Access Manager interface. The breadcrumb navigation is `weblogic > Policies`. The main tabs are `General`, `Resources`, `Default Rules`, `Policies` (selected), and `Delegated Access Admins`. The form fields are as follows:

- Name:** Policy for Broker role
- Description:** Policy for Broker role
- Resource Type:** J2EE\_Role
- Resource:** A list of roles with checkboxes. The `Broker` role is selected.

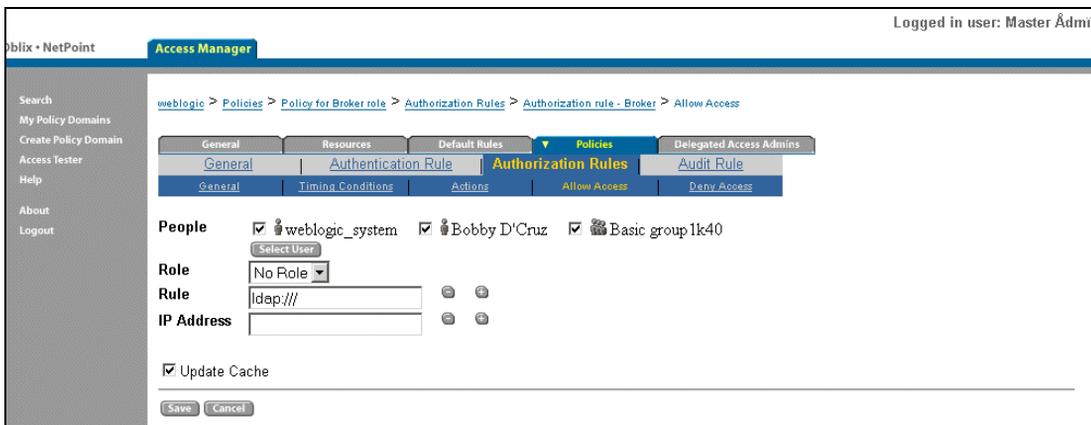
Resource	Description
<input type="radio"/> all	
<input checked="" type="radio"/> J2EE Role	
<input type="checkbox"/> AdminEligible	AdminEligible role for BEA Portal server. Special user group that includes users who are eligible for delegated administration. A user must be placed in this user group before they can be delegated administration tasks.
<input type="checkbox"/> Administrator	Special unchangeable user group called Administrator used to start and stop WebLogic Server. User mapped to system user should be part of this role.
<input checked="" type="checkbox"/> Broker	
<input type="checkbox"/> DelegatedAdministrator	DelegatedAdministrator role for BEA Portal server Includes the logical designations of PortalAdministrator (PA) and Group Administrator (GA). Users in the DelegatedAdministrator user group are in this group for as long as they are either a PA or a GA.
<input type="checkbox"/> SampleAdmin	
<input type="checkbox"/> SystemAdministrator	SystemAdministrator role for BEA Portal server. User group for all-powerful administrators.
<input type="checkbox"/> everyone	everyone group for BEA Weblogic server. This group is Used by BEA Portal server as well. everyone groups contains all users in the system.

Update Cache

## 3. Then add an authorization rule for this J2EE\_Role.



4. Next, allow access to some NetPoint users and NetPoint Groups.

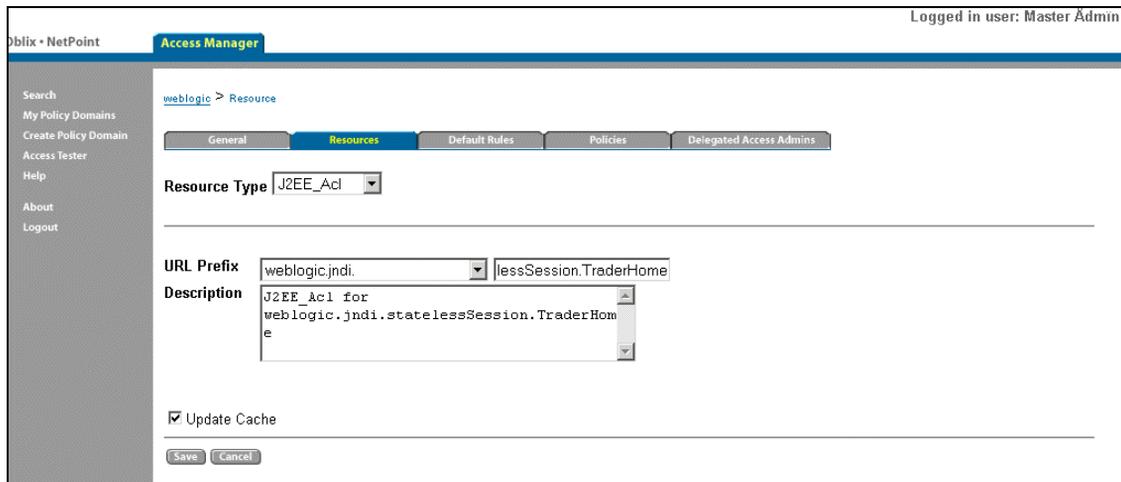


## Example: Defining a J2EE\_Acl to protect an EJB or a WebApp

An EJB can be protected by defining a J2EE\_Acl to protect a resource that the EJB uses. For example an ACL could protect a JNDI name.

### To complete this example

1. In NetPoint you should create a J2EE\_Acl for the JNDI name `weblogic.jndi.statelessSession.TraderHome`.



2. Add a policy for this J2EE\_Acl. While adding the policy for J2EE\_Acl, you can make a pattern such as "\*" to apply the policy to all J2EE\_Acl of type:
  - weblogic.jndi.statelessSession.TraderHome.SomeResource
  - weblogic.jndi.statelessSession.TraderHome.SomeMoreResource
3. Make a pattern such as "~.\*" to apply the policy to all ACL of type:
  - weblogic.jndi.statelessSession.TraderHome.SomeResource.\*
  - weblogic.jndi.statelessSession.TraderHome.SomeResource.More.\*

Logged in user: Master Adm

**Access Manager**

**Resource Operations**

BOOT       SHUTDOWN       LOCKSERVER  
 UNLOCKSERVER       SEND       RECIEVE  
 RESERVE       RESIZE       SHRINK  
 UNLOCKUSER       LOOKUP       MODIFY  
 LIST       EXECUTE       SUBMIT  
 WRITE

**Resource**

all  
 **URL Prefix**      **Description**

<input type="checkbox"/>	weblogic.admin	J2EE_Acl for weblogic.admin
<input type="checkbox"/>	weblogic.event	J2EE_Acl for weblogic.event
<input type="checkbox"/>	weblogic.jdbc.connectionPool	J2EE_Acl for weblogic.jdbc.connectionPool
<input type="checkbox"/>	weblogic.jms.queue	J2EE_Acl for weblogic.jms.queue
<input type="checkbox"/>	weblogic.jms.topic	J2EE_Acl for weblogic.jms.topic
<input type="checkbox"/>	weblogic.jndi	J2EE_Acl for weblogic.jndi
<input type="checkbox"/>	weblogic.jndi.BEA_wls_rce	J2EE_Acl for weblogic.jndi.BEA_wls_rce
<input type="checkbox"/>	weblogic.jndi.java%3Acomp	J2EE_Acl for weblogic.jndi.java.comp
<input type="checkbox"/>	weblogic.jndi.javax	J2EE_Acl for weblogic.jndi.javax
<input type="checkbox"/>	weblogic.jndi.p13nApp	J2EE_Acl for weblogic.jndi.p13nApp
<input type="checkbox"/>	weblogic.jndi.portal	J2EE_Acl for weblogic.jndi.portal
<input checked="" type="checkbox"/>	weblogic.jndi.statelessSession.TraderHome	J2EE_Acl for weblogic.jndi.statelessSession.TraderHome
<input type="checkbox"/>	weblogic.jndi.weblogic	J2EE_Acl for weblogic.jndi.weblogic
<input type="checkbox"/>	weblogic.jndi.wlcsApp	J2EE_Acl for weblogic.jndi.wlcsApp
<input type="checkbox"/>	weblogic.passwordpolicy	J2EE_Acl for weblogic.passwordpolicy
<input type="checkbox"/>	weblogic.server	J2EE_Acl for weblogic.server
<input type="checkbox"/>	weblogic.servlet	J2EE_Acl for weblogic.servlet
<input type="checkbox"/>	weblogic.workspace	J2EE_Acl for weblogic.workspace

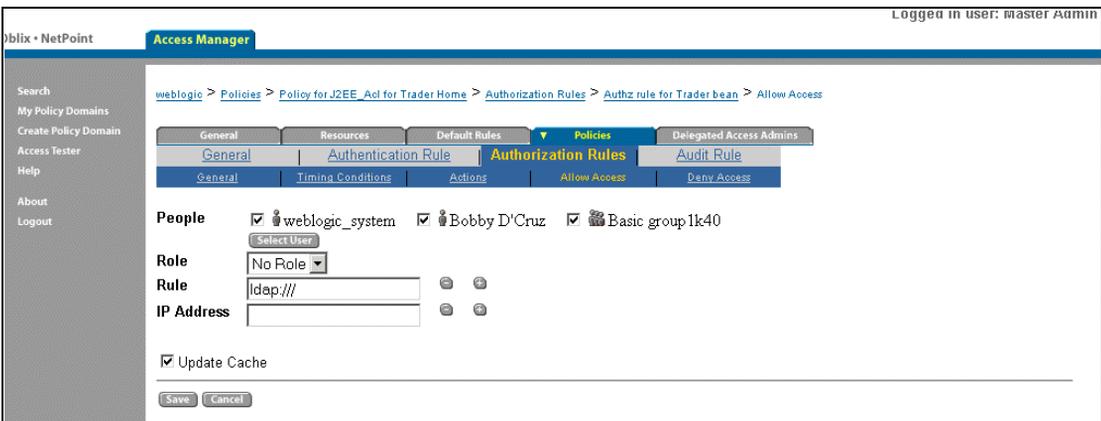
**Pattern**

Update Cache

4. Add an authorization rule for this J2EE\_Acl.



## 5. Allow access to some NetPoint users and NetPoint Groups.



## Advanced Configuration Options

This section describes advanced configuration options that you may want to implement:

- “Changing weblogic\_system User” on page 74
- “Configuring Single Sign-On for the WebLogic Portal” on page 74

## Changing weblogic\_system User

You may want to change the weblogic\_system user that was generated using the NetPoint BEA Ready Realm Configuration tool in the Access System Console.

### To change the weblogic\_system user

1. Determine which NetPoint user will act as the system account for WebLogic.
2. Replace weblogic\_system user with this new user on the Participant screens for each BEA workflow step.  
See “Generating NetPoint Ready Realm for BEA Information” on page 37.
3. Replace weblogic\_system user with this new user on the Set Searchbase and attribute access control screens.  
See “Generating NetPoint Ready Realm for BEA Information” on page 37.
4. Replace the weblogic\_system user in the NetPointBEARealm.properties file with the new user.
5. Replace weblogic\_system user with this new user on the Default Authorization Rule screens for the WebLogic policy domain.
6. Replace weblogic\_system user with this new user as the delegated administrator for the WebLogic policy domain.

## Configuring Single Sign-On for the WebLogic Portal

Configuring single sign-on between the NetPoint Access System and the WebLogic Portal requires some additional configuration in order to send the ObSSOCookie from the browser to the WebLogic Portal.

### Prerequisites

The following issues are important prerequisites for configuring single sign-on:

- NetPoint Ready Realm for BEA must be installed and configured for the WebLogic Portal.
- The WebGateStatic.lst file in *WebGate\_install\_dir/access/oblix/apps/webgate* must have the following line: `IPValidation:false`
- In some situations the Apache Reverse Proxy does not pass the ObSSOCookie to BEA WebLogic after a successful authentication. To avoid this issue, use Form Based authentication instead of Basic Over LDAP when using Apache Reverse Proxy with BEA WebLogic.

## Task overview: Main configuration tasks

1. Install the Login Framework and Oblix classes. The Login Framework is a component of BEA. See “Installing the Login Framework and Oblix Classes” on page 75.
2. In the EBCC application, incorporate Oblix cookie handling into the portal security Web flow. See “Incorporating ObSSOCookie Handling” on page 76.

## Installing the Login Framework and Oblix Classes

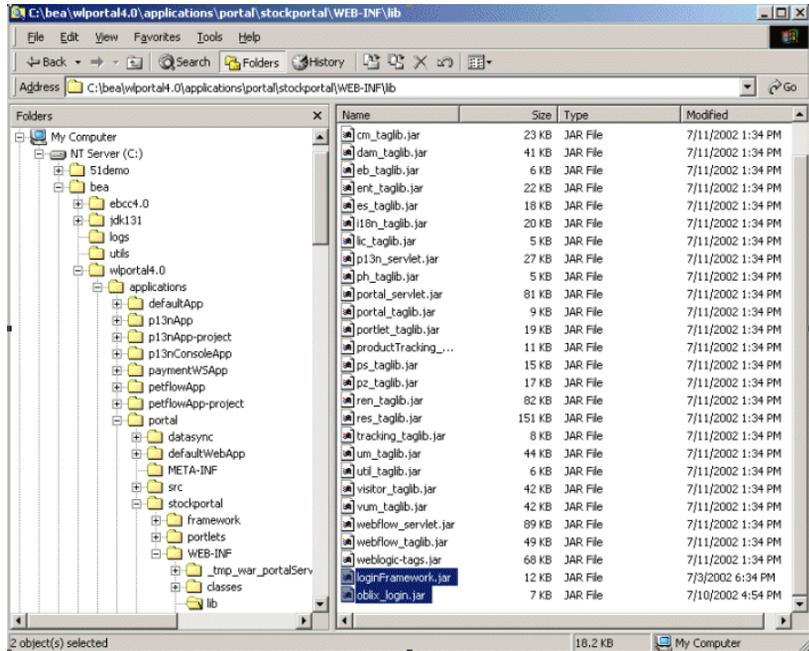
You must install the Login Framework Oblix classes for SSO with WebLogic Portal Server.

The following tasks illustrate installing the Login Framework Oblix classes using the Sample Portal Project.

### To install the Login Framework and Oblix classes

1. Access the Login Framework component from BEA, which is available on BEA’s dev2dev site at <http://dev2dev.bea.com>.
2. Copy the .jar files `loginFramework.jar` and `oblix_login.jar` located in the directory `BEA_install_dir\examples\portalSSO` to your Portal Web applications directory, `WEB-INF\lib`.

where `BEA_install_dir` is the directory where you installed the BEA Ready Realm.



3. Add the LOGIN\_HELPER\_CLASS context parameter with a value of com.oblix.portalSSO.OblixLoginHelper to your web.xml file.

You can add the parameter anywhere between the <web-app> and </web-app> tags in the web.xml file. An example of the parameter is shown below.

```
<context-param>
<param-name>LOGIN_HELPER_CLASS</param-name>
<param-value>com.oblix.portalSSO.OblixLoginHelper</
param-value>
</context-param>
```

The class, com.oblix.portalSSO.OblixLoginHelper, is an implementation of LoginHelper class used by Portal Login Framework to handle single sign-on for users who are logged in to a proxy server such as WebGate.

## Incorporating ObSSOCookie Handling

You must incorporate ObSSOCookie handling into the Portal Security Webflow.

### To incorporate ObSSOCookie Handling

1. In the EBCC, open the Security Webflow.
2. Add a new input processor. Configure the input processor as follows:
  - **Name**—sessionSetupProcessor

- **Class**—com.bea.portal.appflow.processor.security.SessionSetupProcessor
3. Add the following two events to it:
    - **success**—This event must lead to the groupProcessor.
    - **failure**—This event must lead to an error page. This could be the portal login error page or a user-defined error page.
  4. Add a new input processor. Configure the input processor as follows:
    - **Name**—OblixLogOutProcessor
    - **Class**—com.oblix.portalSSO.OblixLogOutProcessor
  5. Modify the link.logout event, which goes to logoutProcessor, so that it goes to OblixLogOutProcessor instead.
  6. Add an event called link.logout, that goes from OblixLogOutProcessor to logoutProcessor.
  7. Open the Portal webflow and add a new input processor.
  8. Configure the input processor as follows:
    - **Name**—implicitLoginProcessor
    - **Class**—com.bea.portal.appflow.processor.security.ImplicitLoginProcessor
  9. Make implicitLoginProcessor the begin node of the portal webflow:
    - a) Click the Begin Node icon.  
The Set the Begin Node window appears.
    - b) Select implicitLoginProcessor from the drop-down list.
  10. In the Portal Webflow, add an input node called sessionSetupProcessor:
    - a) Click the Proxy Node icon to create a proxy node.
    - b) Click the value field of referent-namespace to view the available web flows under Sample Portal.
    - c) Select Security to specify the Security webflow.
    - d) In the value field for entity-name, select sessionSetupProcessor.  
The proxy node is now named sessionSetupProcessor.
  11. Add the following three events to implicitLoginProcessor:
    - a) bea.portal.framework.internal.setupSession.  
This event must lead to the security namespace's sessionSetupProcessor.
    - b) bea.portal.framework.internal.noSetupSession  
This must lead to the preProcessor input processor.

c) `bea.portal.framework.internal.WebLogicLoginFailed`

This exception gets thrown if the `implicitLoginProcessor` fails in its attempts to log the user into WebLogic Server. The event must lead to the appropriate error handling page, or to the `preProcessor` input processor for the user to be allowed anonymous access to the Portal on a failed login.

12. Incorporate the `ObSSOCookie` processor into the webflow as described below.

---

**Note:** In Weblogic Portal 4.0, you must incorporate the `ObSSOCookie` processor into the portal webflow; in Weblogic Portal 7.0, you must incorporate the `ObSSO` cookie processor into the security webflow.

---

### **To incorporate the `ObSSOCookie` processor into the portal webflow in WLP 4.0**

1. Add a new input processor that is configured as follows:
  - **Name**—`OblixCookieProcessor`
  - **Class**—`com.oblix.portalSSO.OblixCookieProcessor`
2. Modify the event `bea.portal.framework.internal.postLogin` by pointing it from `preProcessor` to `OblixCookieProcessor`.
3. Add an event `bea.portal.framework.internal.postLogin` from the `OblixCookieProcessor` to the `security_groupProcessor`.
4. Select the security webflow and click Save.
5. Select the portal webflow and click Save.
6. Synchronize to the server.
7. Restart the Portal server.

### **To incorporate the `ObSSOCookie` processor into the security webflow in WLP 7.0**

1. Add a new input processor to the security webflow that is configured as follows:
  - **Name**—`OblixCookieProcessor_WLP7`.
  - **Class**—`com.oblix.portalSSO.OblixCookieProcessor_WLP7`.
2. Modify the event named `success` pointing from `groupProcessor` to `user_account_postLoginProcessor` as follows:

Disconnect it from the `user_account_postLoginProcessor` and connect it to `OblixCookieProcessor_WLP7` instead.
3. Add an event named `success` from `OblixCookieProcessor_WLP7` to `user_account_postLoginProcessor`.

4. Select the security webflow and click Save.
5. Save the portal and synchronize to the server.
6. Restart the Portal server.

## Changes to Workflow

The NetPointBEARealm uses workflows defined in the NetPoint COREid system to create and delete users. The data available to be passed in a workflow request is limited by realm interface of WebLogic. The Userid and Password parameters are available to the realm while making a Create User request. The only attribute that is available while deleting a user is Userid. It is possible to define a workflow that uses values for these two attributes. It is also possible to send constant values for more attributes as shown, in the following sample workflow definition.

**Table 3** Workflow Fields and Values

Workflow Field	Sample Workflow Values
Workflow Name	Name generated by the NetPoint BEA Realm Create User workflow
Workflow Type:	Create User
Workflow DN:	obworkflowid=wfqs20020806T0907402920,obcontainerId=workflowDefinitions,OU=Oblix,OU=Company,DC=qalab-vduong,DC=oblix,DC=com
Workflow Status:	Enabled
Description:	Workflow generated for NetPointBEARealm
Target:	Company:OU=Company,DC=qalab-vduong,DC=oblix,DC=com
Workflow Domain:	OU=Company,DC=qalab-vduong,DC=oblix,DC=com
Workflow Steps	<p>Step 1:            Name: Initiate            Attribute Name: LoginID (Required)            Attribute Name: Password (Required)            Attribute Name: Name (Required)            Participant: weblogic_system</p> <p>Step 2:            Name: Enable            Entry Condition:            1. true:false</p>

The following are the corresponding parameters from NetPointBEARealm.properties file:

```
OBWebPass.CreatUserWorkFlowID= wfqs20020806T0907402920,  
obcontainerId=workflowDefinitions, OU=Oblix,  
OU=Company,DC=qalab-vduong,DC=obl ix,DC=com  
OBWebPass.CreatUserWorkFlowDomain=  
OU=Company,DC=qalab-vduong,DC=obl ix,DC=com:
```

\$UID\$ and \$PASSWORD\$ denote value of login attribute and password, respectively, in this file. The placeholders will be passed to the workflow as-is and will be written to the user profile. Both \$UID\$ and \$PASSWORD\$ will be replaced with values obtained at runtime for the login attribute and password.

```
OBWebPass.CreatUserWorkFlowNumOfFields=3  
OBWebPass.CreatUserWorkFlowAttrName_1=cn  
OBWebPass.CreatUserWorkFlowAttrValue_1=Name of $UID$  
OBWebPass.CreatUserWorkFlowAttrName_2=uid  
OBWebPass.CreatUserWorkFlowAttrValue_2=$UID$  
OBWebPass.CreatUserWorkFlowAttrName_3=userPassword  
OBWebPass.CreatUserWorkFlowAttrValue_3=$PASSWORD$  
OBWebPass.CreatUserWorkFlowComment=Added user $UID$ from weblogic  
portal server
```

If the workflow is modified to use different attributes, the preceding sample lines in the NetPointBEARealm.properties file need to change. If workflow is modified to use another attribute, the DS attribute name will have to be specified in this file. For example, if you change the first attribute from cn Name to cn Mail (that is obmail in DS) then do the following:

```
OBWebPass.CreatUserWorkFlowAttrName_1=obmail  
OBWebPass.CreatUserWorkFlowAttrValue_1= $UID$@company.com
```

Similarly, if this is a new parameter being added to the workflow, you need to increase the number of fields and add two new lines for attribute:

```
OBWebPass.CreatUserWorkFlowNumOfFields=4
```

And add two new lines as below:

```
OBWebPass.CreatUserWorkFlowAttrName_4=obmail  
OBWebPass.CreatUserWorkFlowAttrValue_4=$UID$@company.com
```

# Samples for NetPoint Ready Realm for BEA

This section contains examples to illustrate features and functions of NetPoint Ready Realm for BEA. Unless otherwise specified, these samples are located in

*install\_dir*/examples

where *install\_dir* is the directory where the BEA Ready Realm is installed.

For example:

c:\NetPoint BEA Ready Realm\examples

The samples are:

- Single Sign-On (SSO) with Form login (WebLogic 6.x)
- Roles in Deployment Descriptor
- Single Sign-on (SSO) Servlet
- Mapping Roles in Deployment Descriptor to NetPoint Roles
- WebLogic Dynamic Role Mapping Example
- Custom ACL

## Single Sign-On (SSO) with Form Login

This sample demonstrates how NetPoint Ready Realm for BEA allows single sign-on (SSO) for Web resources protected with WebLogic 6.x.

### To build a security.war file

1. Change the directory to *install\_dir*/examples/webapps/security where *install\_dir* is the directory where you installed BEA Ready Realm.
2. Modify the build.sh (Unix) or build.cmd (Windows) script to match your environment
3. Execute build.sh (Unix) or build.cmd (Windows) to build the security.war file

### To deploy the WAR file

1. On WebLogic 5.x, add the following line to the weblogic.properties file:  
weblogic.httpd.webApp.security=%WL\_Home%/myserver/security.war
2. Restart the WebLogic server.

---

**Note:** For 6.x or 7.x, refer to BEA's documentation on deploying WAR's and EJB's.

---

## **To set up NetPoint**

1. Set up the policy for J2EE\_Role SampleAdmin.
2. Set up the authorization rule for J2EE\_Role SampleAdmin.  
See “Protecting Resources With Policy Domains” on page 65 for details.

## **To run the sample**

1. To run the sample, set up the single sign-on from WebLogic to WebGate:  
Navigate to `http://server.mydomain.com:7001/security`.  
A link to the login page appears.
2. Click the link and enter your username and password in the login form that appears.

For an authorized user in the SampleAdmin role, you should see a page that allows you to change the background color of the page. If you run the example with an unauthorized user ID, you should see an appropriate error message.

After you authenticate to WebLogic, you should be able to go to a resource protected by WebGate and access it without having to authenticate with WebGate.

## **To set up SSO from WebGate to WebLogic**

1. Open a browser and access a resource protected by WebGate.
2. Authenticate to WebGate and obtain an ObSSOCookie.
3. Navigate to `http://server.mydomain.com:7001/security`.

The page should tell the current user is logged in, and if this user is in the SampleAdmin role, you can click on the link to change the background color.

If the user is not in the SampleAdmin role, he should still see his name on the welcome page. However, he will be directed to the login page upon clicking the “Change Background Color” link. This happens because even though the user is authenticated to WebLogic by SSO, he does not belong to the role SampleAdmin and therefore he is not authorized to perform that operation.

## Roles in Deployment Descriptor

This program demonstrates how roles defined in the Deployment Descriptor are honored. The `ejb-jar.xml` file defines a Broker role. This example maps the Broker role to users J.Smith and L.Reed, who are users defined in NetPoint. It has four procedures that you perform in the order listed.

### To build a sample

1. Change the directory to `install_dir/examples/ejb/basic/statelessSession`
2. Modify the `build.sh` (Unix) or `build.cmd` (Windows) script to match your environment.
3. Execute `build.sh` (Unix) or `build.cmd` (Windows) to build the sample.

### To set up WebLogic

1. On WebLogic 5.x, deploy the EJB by making appropriate changes to the `weblogic.properties` file.
2. Restart the WebLogic server.

---

**Note:** To deploy on 6.x or 7.x, refer to BEA's documentation on deploying WARs and EJBs.

---

### To set up NetPoint

1. Ensure that the J2EERole Broker is defined in the NetPoint Access System and that users J.Smith and L.Reed are authorized by an authorization rule for the role.
2. Use the access tester to ensure that users J.Smith and L.Reed have access to `J2EE_Acl weblogic.jndi.statelessSession.TraderHome`. If not, modify the authorization rule for the appropriate J2EE\_Acl to allow lookup operation for these users.

### To run a sample

1. Run the sample client with the following command:  

```
$ java examples.ejb.basic.statelessSession.Client t3://localhost:7001 <NetPoint UserID> <NetPoint User Password>
```

For example:

```
$ java examples.ejb.basic.statelessSession.Client t3://localhost:7001 J.Smith J.Smith
```

If you run the example with an unauthorized user's ID, you will see an error message.

## Single Sign-On (SSO) Servlet

This program demonstrates how NetPoint Ready Realm for BEA functions can be used to create user session from the ObSSOCookie that was obtained from a WebGate. And, conversely, it can generate and set an ObSSOCookie after a user is logged in using NetPoint Ready Realm for BEA.

### To build a sample

1. Change the directory to `install_dir/examples/servlet`  
where `install_dir` is the directory where you installed BEA Ready Realm.
2. Modify the makefile (Unix) or build.cmd (Windows) script to match your environment
3. Execute makefile (Unix) or build.cmd (Windows) to build the sample

### To set up WebLogic

1. For WebLogic 5.1, register the servlet by inserting the following line in the `weblogic.properties` file:

```
weblogic.httpd.register.SSOServlet=obaccess.SSOServlet
```

For WebLogic 6.x or 7.x, deploy the servlet by following the instructions at [http://e-docs.bea.com/wls/docs61/quickstart/quick\\_start.html](http://e-docs.bea.com/wls/docs61/quickstart/quick_start.html), Servlet Quick Start.

2. Restart the WebLogic server
3. Using the WebLogic Console, verify that servlet `SSOServlet` has been successfully deployed.

### To set up NetPoint (applicable only to WebLogic 5.1)

1. Verify the following resource is defined in the NetPoint Access System.

Resource Type	weblogic_url
Resource URL	/

2. Create a policy to protect the resource `/index.html` of type `weblogic_url`.
3. Authorize at least one person to access the resource.  
In this example, J. Smith is used.
4. Use the Access Tester to verify that J.Smith indeed has access to `/index.html` user operation `ACCESS` and resource type `weblogic_url`.

## To run the sample

1. Navigate to the following URL:

```
http://WebLogic_host:port/SSOServlet?REQUEST=/  
index.html&USERNAME=J.Smith&PASSWORD=J.Smith
```

The first time this URL is accessed, the following message appears in the browser client:

```
J.Smith is "cn=John Smith,ou=Corporate,o=Company,c=US"
```

```
J.Smith@NetPointBEAReadyRealm authorized. You will be redirected  
to the requested page in 5 seconds.
```

This means that J.Smith has successfully logged in. On subsequent hits to the same URL, you see the following message:

```
User "cn=John Smith,ou=Corporate,o=Company,c=US" already logged  
in.
```

```
Session start time: 992387653
```

```
Session last use time: 992387754
```

```
Authenticated level: 1
```

This means that J.Smith is already logged in. Information about the session is then displayed.

You could also get to a protected resource in WebGate and get an ObSSOCookie. Then using the same browser go to the SSOServlet. It will sign the user in to WebLogic and generate the page with session information as above.

## Mapping EJB Roles to NetPoint Roles

This program demonstrates mapping of EJB roles to NetPoint roles. The deployment descriptor contains roles GenBull, GenBear, and GenBigBrother defined in the `ejb-jar.xml` file. In the file `weblogix-ejb-jar.xml` these roles are mapped to Bull, Bear, and BigBrother, respectively, which are roles defined in NetPoint.

### To build a sample

1. Change the directory to `install_dir/examples/ejb/basic/statefulSession`.
2. Modify the `build.sh` (Unix) or `build.cmd` (Windows) script to match your environment.
3. Execute `build.sh` (Unix) or `build.cmd` (Windows) to build the sample.

### To set up WebLogic

1. Deploy the EJB by making appropriate changes to the `weblogic.properties` file.
2. Restart the WebLogic server.

## To set up NetPoint

1. Be sure that J2EE\_Role Bull, Bear, and BigBrother are set up in NetPoint.
2. Use the Access Tester to verify that J.Smith has been assigned to the BigBrother role, L.Reed to Bear, and A.Collins to Bull, respectively, by testing access of each user for respective J2EE\_Role.

## To run a sample

1. Run the sample client with the following command:

```
$ java examples.ejb.basic.statefulSession.Client t3://  
localhost:7001 <NetPoint UserID> <NetPoint User Password>
```

For example:

```
$ java examples.ejb.basic.statefulSession.Client t3://  
localhost:7001 J.Smith J.Smith
```

For an authorized user in an appropriate role you should see the results. If you run the example with an unauthorized user's ID, you should see an appropriate error message.

## WebLogic Dynamic Role Mapping

This program demonstrates how WebLogic SSO works with NetPoint Ready Realm for BEA and how the roles defined in deployment descriptor of the bean are honored.

The DynamicRoleMapServlet uses the statefulSession bean described in previous sample. It has to be deployed and working before this sample can be executed.

## To build a sample

1. Change directory to *install\_dir/examples/servlet*.
2. Compile the DynamicRoleMapServlet as follows:

```
$ javac -d %SERVLET_CLASSES% DynamicRoleMapServlet.java
```

## To set up WebLogic

1. For WebLogic 5.1, deploy the servlet by adding the following line to the weblogic.properties file:

```
weblogic.httpd.register.DynamicRoleMapServlet=  
examples.servlet.DynamicRoleMapServlet
```

For WebLogic 6.x, deploy the servlet by following the instructions available at [http://e-docs.bea.com/wls/docs61/quickstart/quick\\_start.html](http://e-docs.bea.com/wls/docs61/quickstart/quick_start.html).

2. Ensure the statefulSession bean mentioned in “Mapping EJB Roles to NetPoint Roles” on page 85 is deployed.
3. Restart the WebLogic server.

## To set up NetPoint

1. For 6.x make sure the servlet is protected by modifying the web.xml file in the webapp.
2. Create a policy to protect the resource /DynamicRoleMapServlet of type weblogic\_url.
3. Authorize at least one person to access the resource.  
This example uses J. Smith.
4. Use the Access Tester to verify that J.Smith has access to /DynamicRoleMapServlet user operation ACCESS and resource type weblogic\_url.

## To run a sample

Navigate to `http://localhost:7001/DynamicRoleMapServlet`.

You should be prompted for a user name and password by the browser or redirected to login form depending. Log in as J. Smith. Whether you are prompted or redirected depends on the setup in WebLogic.

For an authorized user in an appropriate role you should see the results. If you run the example with an unauthorized user's ID, you should see an appropriate error message.

## Custom ACL

This program shows how a programmatic ACL check can be used. The sample uses the standard WebLogic sample *Frob* to demonstrate how NetPoint Ready Realm for BEA restricts access to an WebLogic object, such as an RMI object, by setting up an ACL to be protected in NetPoint.

### To building the sample

1. Change directory to `%WL_Home%/examples/security/acl`.
2. Open `Package-examples.security.acl.html` and follow the instructions in this file to build the sample.

---

**Note:** While modifying the `weblogic.properties` file, skip the steps for adding *joouser* and adding the ACL *aclexample*. These will be configured in NetPoint.

---

### To set up WebLogic

1. Ensure that the `FrobImpl` instance has been registered as an RMI Bea class as described in `Package-examples.security.acl.html`.
2. Restart the WebLogic server.

## To set up NetPoint

1. Define a J2EE\_Acl with URL prefix, acl example.
2. Add a resource operation FROB to the J2EE ACL resource type in the NetPoint System Console.
3. Create a policy for J2EE\_Acl example for operation FROB.
4. Authorize at least one person in authorization rule for this policy.  
In this example, J. Smith is used.
5. Use the Access Tester to verify that J. Smith has access to J2EE\_Acl aclexample.

## To run a sample

1. Run the sample client with the following command:

```
$ java examples.security.acl.Client t3://localhost:7001 <NetPoint UserID>  
<NetPoint User Password>
```

For example:

```
$ java examples.security.acl.Client t3://localhost:7001 J.Smith J.Smith
```

2. In response to an authorized user for J2EE\_Acl aclexample in NetPoint, you should see the result Frobbed Successfully. If you run the example with an unauthorized user's ID, you should see the message Failed to frob displayed.

# Implementation Notes for Active Directory

Consider the following issues when implementing NetPoint Ready Realm for BEA on Active Directory.

## NetPoint Ready Realm for BEA Configuration Fails

If you are running Active Directory using multiple domains or an Active Directory forest, the automatic policy creation for NetPoint Ready Realm for BEA fails after adding the weblogic\_system user.

During the first step of the automatic configuration, the weblogic\_system user is added, and then weblogic policies are added. The policies refer to the weblogic\_system user. However, there is some delay (anywhere from 5 minutes to 15 minutes) in Active Directory's internal cache update. Therefore, when Active Directory does a referential integrity check on the policies, it fails because it cannot locate the recently created weblogic\_system user.

If you will see the following error:

```
"Error in adding meta data needed for the web Resource Management  
console"
```

You will have to go to Access Manager and delete the partially generated WebLogic policy domain. After waiting for the Active Directory caches to synchronize, you must rerun the setup.

During the first step of the NetPoint Ready Realm for BEA configuration, wait for a few minutes and click the NetPoint BEA Ready Realm Configuration button of the Access System Console again. If you still see the same error, try again after some time. Automatic policy generation will be successful once the internal caches are updated.

## Create User Fails Talking to Active Directory Forest

If you are running Active Directory using multiple domains or an Active Directory Forest, the create user workflow is invoked from an IdentityXML call. There will be a delay between the time that the create user operation succeeds and the time that the user is synched to global catalog of Active Directory. When the WebLogic server initially looks for the user realm, it queries the global catalog and may not find the user. The user will be seen in the realm as soon as the Active Directory global catalog is updated.

## Set Active Directory in NetPointBEARealm.properties

If you are running Active Directory using multiple domains, you must manually edit the NetPointBEARealm.properties file to include a value for the OBWebPass.ADDomain parameter.

For example, OBWebPass.ADDomain=dc=goodwill, dc=oblix, dc=com

## Set the User Mapping in NetPointBEARealm.properties

Specify the domain and userid with the slash (/) in escaped format (that is, domain\\userid) in an Active Directory forest environment.

For example, in a non-Active Directory Forest environment:

```
OBSystemUser=JSmith
```

or

```
OBGuestUser=JSmith
```

In an Active Directory Forest environment:

```
OBSystemUser=goodwill\\JSmith
```

or

```
OBGuestUser=goodwill\\JSmith
```

where *goodwill* is the value of the domain used for logging into WebGate when using Active Directory forest.

## Set Authentication Scheme for Active Directory

When you select BEA Ready Realm configuration to generate policies for J2EE\_Roles and J2EE\_ACLS, two authentication schemes are generated. One authentication scheme is for a standard Active Directory and the other is for an Active Directory forest.

If the installation is using an Active Directory forest, use the Active Directory Forest authentication scheme.

### To use the Active Directory Forest authentication scheme

1. Modify the default authentication rule for weblogic policy domain to use the Authentication scheme for Active Directory forest.
2. Modify the authentication rules specified in policy for /Authen/Basic resource to use the Authentication scheme for the Active Directory forest. This is the first policy in the WebLogic policy domain.

## Troubleshooting

**Problem**—Single slashes are not recognized by Ready Realm for BEA.

**Solution**—Append the NetPointBEARealm.properties file. For Windows and edit the OBAccessSDK.InstallDir parameter. Make sure that there are double slashes in the path; for example:

OBAccessSDK.InstallDir=D:\\NetPoint\\NetPointBEARealm). This is a known issue for Windows in this release.

**Problem**—I get “All the jars are not in classpath – NoClassDefException”

**Solution**—Make sure that the HTTPClient.jar and its patch are in the classpath.

**Problem**—I get an “SSLPeerUnverifiedException - peer not authenticated” exception.

**Solution**—The jvm being used is different from the jvm that has imported the certificates of ca and server. The jvm and keytools used must be from the same installation. If one keytool is used to add certificates and java is invoked from the other installation directory, the jvm will not be able to use the certificates and will produce this exception.

**Problem**—I get an ObConfig.NO\_CONFIG\_FILE message.

**Solution**—This error means that the Access SDK client configuration file was not found. Check to ensure that the `OBAccessSDK.InstallDir` points to the Access SDK installation directory. If you are using NetPoint 6.0 or later versions, then the Access SDK is bundled with NetPoint BEARealm. From Release 6.0 onwards, `OBAccessSDK.InstallDir` should point to the NetPoint BEA Realm installation directory; for example,

```
OBAccessSDK.InstallDir=c:\\NetPoint\\NetPointBEARealm
```

**Problem**—I get an `UnsatisfiedLinkError`.

**Solution**—You may not have the Access SDK lib in the `PATH` or `LD_LIBRARY_PATH` depending on the platform; for example on NT set:

```
set PATH=%PATH%;c:\\NetPoint\\NetPointBEARealm\\oblix\\lib
```

For **Solaris**, set:

```
setenv LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/NetPoint/  
NetPointBEARealm/oblix/lib export LD_LIBRARY_PATH
```

This can be either done at system level or at start-up.

**Problem**—I get `NoClassDefFound` error for `com.oblix.access.ObAccessException` or `com.oblix.realm.NetPointBEARealm` or `com.oblix.accessmgr.ObAMException`?

See the following error message:

```
The webLogic server did not start up properly.  
Exception raised: java.lang.reflect.InvocationTargetException  
java.lang.reflect.InvocationTargetException:  
java.lang.NoClassDefFoundError: com/oblix/accessmgr/  
ObAMException  
at java.lang.Class.forName0(Native Method)  
at java.lang.Class.forName(Class.java:120)  
at weblogic.security.acl.Realm.getRealm(Realm.java:78)  
at weblogic.security.acl.Realm.getRealm(Realm.java:56)  
at weblogic.t3.srvr.T3Srvr.initializeSecurity(T3Srvr.java:1910)  
at weblogic.t3.srvr.T3Srvr.start(T3Srvr.java:1151)  
at weblogic.t3.srvr.T3Srvr.main(T3Srvr.java:879)  
at java.lang.reflect.Method.invoke(Native Method)  
at weblogic.Server.startServerDynamically(Server.java:140)  
at weblogic.Server.main(Server.java:97)  
at weblogic.Server.main(Server.java:58)  
java.lang.NoClassDefFoundError: com/oblix/accessmgr/  
ObAMException  
at java.lang.Class.forName0(Native Method)
```

```
at java.lang.Class.forName(Class.java:120)
at weblogic.security.acl.Realm.getRealm(Realm.java:78)
at weblogic.security.acl.Realm.getRealm(Realm.java:56)
at weblogic.t3.srvr.T3Srvr.initializeSecurity(T3Srvr.java:1910)
at weblogic.t3.srvr.T3Srvr.start(T3Srvr.java:1151)
at weblogic.t3.srvr.T3Srvr.main(T3Srvr.java:879)
at java.lang.reflect.Method.invoke(Native Method)
at weblogic.Server.startServerDynamically(Server.java:140)
at weblogic.Server.main(Server.java:97)
at weblogic.Server.main(Server.java:58)
```

**Solution**—You probably have the CLASSPATH wrong. Check the documentation for setting this variable.

**Problem**—I get InvalidCredential error when I boot the WebLogic Server or WebLogic Portal and provide the credential at boot time.

**Solution**—InvalidCredential error means that the password is not correct. This can happen for the following reasons:

- If the WebLogic Server is not yet configured to use the real password, then the password provided at boot time must be the one that was provided while installing the WebLogic server.
- After the NetPoint Ready Realm for BEA is configured and WebLogic Server is configured to use the realm, the server password provided at boot time has to be for the user who is mapped to the system user in NetPointBEARealm.properties file. If you are using the default weblogic\_system user that was created by automatic configuration, then make sure that the password is set for this user by setting it from NetPoint User Manager. By default this password is not set with automatic configuration.
- After the NetPoint Ready Realm for BEA is configured and WebLogic Server is configured to use the realm, the password provided at boot time has to be that of the user that is mapped to the system user in NetPointBEARealm.properties file. For example, If you have a user mapped to the system user, for instance, OBSYSTEMUSER=JohnSmith, you must provide the password for John Smith when WebLogic boots up.

**Problem**—I get an “/authn/basic is not protected” error on the WebLogic console. Also, WebLogic does not boot.

**Solution**—This happens if automatic configuration was used to generate WebLogic policies, but the Access Server was not re-started after the policies were created. The automatic creation adds new resource types that the Access Server needs to pick up.

This may also happen if the ACL entries in the `filerealm.properties` file were not commented out.

**Problem**—I see a blank page on Portal Servers stockportal example and a `NullPointerException` on the WebLogic console.

**Solution**—WebLogic Portal Server needs a user called `guest` to be in the directory. If you do not have one, create a user with login attribute `guest`.

**Problem**—I have removed all the ACLs from `fileRealm.properties` file and now my WebLogic Server does not boot.

**Solution**—This happens if the ACLs are removed from `fileRealm.properties` file and the NetPoint Ready Realm for BEA is not yet configured to work. WebLogic will be unable to get ACLs from the NetPoint Ready Realm for BEA or from `filerealm`, and will not be able to boot. To remedy the situation:

1. Un-comment the `fileRealm` ACLs.
2. Boot the server.
3. Configure it to use `NetPointBEARealm`.  
See “Configuring WebLogic for Ready Realm” on page 52 for more information.
4. Restart the server.
5. After you are sure that NetPoint Ready Realm for BEA is working, stop the server and comment out entries from `fileRealm.properties`.
6. Restart the server.

**Problem**—In WebLogic 5.1, the WebLogic server does not start up and I get `java.security.AccessControlException` (detailed exception):

The webLogic server did not start up properly.

```
Exception raised: java.lang.reflect.InvocationTargetException
java.lang.reflect.InvocationTargetException:
java.lang.ExceptionInInitializerErr
or: java.security.AccessControlException: access denied
(java.security.SecurityP
ermission getProperty.ssl.SocketFactory.provider)
at java.security.AccessControlContext.checkPermission
(AccessControlContext.java:195)
at java.security.AccessController.checkPermission
(AccessController.java:403)
at java.lang.SecurityManager.checkPermission
(SecurityManager.java:549)
at java.security.Security.getProperty(Security.java:695)
at javax.net.ssl.SSLSocketFactory$1.run([DashoPro-v1.2-120198])
```

```
at java.security.AccessController.doPrivileged(Native Method)
at javax.net.ssl.SSLSocketFactory.a([DashoPro-v1.2-120198])
at javax.net.ssl.SSLSocketFactory.getDefault
([DashoPro-v1.2-120198])
at HTTPClient.HTTPConnection.<clinit>(HTTPConnection.java:324)
at HTTPClient.AuthorizationInfo.<clinit>
(AuthorizationInfo.java:100)
at com.oblix.soapclient.SoapClient.<init>(SoapClient.java:133)
at com.oblix.realm.NetPointBEARealm.init
(NetPointBEARealm.java:445)
at weblogic.security.acl.Realm.getRealm(Realm.java:79)
at weblogic.security.acl.Realm.getRealm(Realm.java:56)
at weblogic.t3.srvr.T3Srvr.initializeSecurity(T3Srvr.java:1756)
at weblogic.t3.srvr.T3Srvr.start(T3Srvr.java, Compiled Code)
at weblogic.t3.srvr.T3Srvr.main(T3Srvr.java:827)
at java.lang.reflect.Method.invoke(Native Method)
at weblogic.Server.startServerDynamically(Server.java:99)
at weblogic.Server.main(Server.java:65)
at weblogic.Server.main(Server.java:55)
java.lang.ExceptionInInitializerError:
java.security.AccessControlException: acc
ess denied (java.security.SecurityPermission
getProperty.ssl.SocketFactory.provider )
at java.security.AccessControlContext.checkPermission
(AccessControlContext.java:195)
at java.security.AccessController.checkPermission
(AccessController.java:403)
at java.lang.SecurityManager.checkPermission
(SecurityManager.java:549)
at java.security.Security.getProperty(Security.java:695)
at javax.net.ssl.SSLSocketFactory$1.run([DashoPro-v1.2-120198])
at java.security.AccessController.doPrivileged(Native Method)
at javax.net.ssl.SSLSocketFactory.a([DashoPro-v1.2-120198])
at javax.net.ssl.SSLSocketFactory.getDefault
([DashoPro-v1.2-120198])
at HTTPClient.HTTPConnection.<clinit>(HTTPConnection.java:324)
at HTTPClient.AuthorizationInfo.<clinit>
(AuthorizationInfo.java:100)
at com.oblix.soapclient.SoapClient.<init>(SoapClient.java:133)
at com.oblix.realm.NetPointBEARealm.init
(NetPointBEARealm.java:445)
```

```
at weblogic.security.acl.Realm.getRealm(Realm.java:79)
at weblogic.security.acl.Realm.getRealm(Realm.java:56)
at weblogic.t3.srvr.T3Srvr.initializeSecurity(T3Srvr.java:1756)
at weblogic.t3.srvr.T3Srvr.start(T3Srvr.java, Compiled Code)
at weblogic.t3.srvr.T3Srvr.main(T3Srvr.java:827)
at java.lang.reflect.Method.invoke(Native Method)
at weblogic.Server.startServerDynamically(Server.java:99)
at weblogic.Server.main(Server.java:65)
at weblogic.Server.main(Server.java:55)
```

**Solution**—Modify the `weblogic.policy` file and add the following to this file:

```
grant {
    permission java.security.SecurityPermission
    "getProperty.ssl.SocketFactory.provider";
};
```

**Problem**—I not see `ObSSOCookie` being set.

**Solution**

1. Make sure that you are using fully qualified domain names to access the WebLogic Server and the Web server that is running WebGate.

For example, use:

```
http://server1.mycompany.com:7001
```

Not `http://server1:7001`

2. Check the cookie domain:
  - a) Is the Primary Cookie Domain set in the WebGate configuration in Access Manager?
  - b) Is the `OBWebPass.cookieDomain` set in the `NetPointBEARealm.properties` file for WebLogic?

**Problem**—I see the `ObSSOCookie` but WebLogic Server is rejecting it.

**Solution**

- Make sure that the time is synchronized on the machine running WebGate and WebLogic Server
- Make sure that in the authentication schemes for WebGate and the WebLogic server resources have the same level.

**Problem**—I get the `ObSSOCookie`, but WebGate is rejecting it.

**Solution**

- Make sure that in WebGateStatic.lst file in \$WEB\_GATE\_INSTALL\_DIR/access/oblix/apps/webgate contains the line IPValidation:false
- Make sure that the time is synchronized on the machines running WebGate and the WebLogic Server.
- Make sure that in the authentication scheme for WebGate and WebLogic Server resources have the same level.

**Problem**—Using IdentityXML to create a new user or delete an existing user fails, producing an unauthorized exception.

**Solution**

- Make sure that the WebPass host name is fully qualified host name.
- Make sure that user mapped to system user is a participant in workflow to create or delete user.

**Problem**—I see messages similar to those below in my WebLogic logs.

```
<Jul 12, 2002 2:45:22 PM GMT-08:00> <Info> <Security> <Access
failed (Thread = Thread[main,5,main]), caused
java.lang.SecurityException: User "system@NetPointBEARealm"
does not have Permission "modify" based on ACL
"weblogic.jndi.weblogic".>
```

**Solution**—This log message says that the system user does not have permission to modify JNDI name weblogic.jndi.weblogic. If you want to open access to this jndi name in WebLogic to a certain set of users, log in to Access Manager. Add a J2EE\_Acl for weblogic.jndi.weblogic. Add a policy for this J2EE\_Acl and choose MODIFY as the operation for the policy. Then, add an authorization rule for the policy and define the set of users you want to give access too this JNDI name.

**Problem**—I see the following:

```
<Jul 31, 2002 12:21:35 PM PDT> <Error> <NetPointBEARealm> <Fatal
exception while initializing realm null
```

```
Invalid Access Management message: code. Message: To be parsed:
at com.oblix.accessmgr.ObAMSerializedObject.deserializeInt
(ObAMSerializedObject.java:285)
at com.oblix.accessmgr.ObAMException.<init>
(ObAMException.java:342)
at com.oblix.accessmgr.ObAccessManager.sendRequest
(ObAccessManager.java:163)
at com.oblix.accessmgr.ObAccessManager.setAdmin
(ObAccessManager.java:189)
at com.oblix.realm.NetPointBEARealm.init
(NetPointBEARealm.java:479)
at weblogic.security.acl.Realm.getRealm(Realm.java:85)
at weblogic.security.acl.Realm.getRealm(Realm.java:62)
```

```

at weblogic.security.SecurityService.initializeRealm
(SecurityService.java:258)
at weblogic.security.SecurityService.initialize
(SecurityService.java:115)
at weblogic.t3.srvr.T3Srvr.initialize(T3Srvr.java:402)
at weblogic.t3.srvr.T3Srvr.run(T3Srvr.java:202)
at weblogic.Server.main(Server.java:35)
cc
<Jul 31, 2002 12:21:35 PM PDT> <Emergency> <Server> <Unable to
initialize the server: 'Fatal initialization exception
Throwable: java.lang.IllegalAccessError:
com.oblix.realm.NetPointBEARealm$3: Fatal exception while
initializing realm
java.lang.IllegalAccessError:
com.oblix.realm.NetPointBEARealm$3: Fatal exception while
initializing realm
at weblogic.security.acl.Realm.getRealm(Realm.java:91)
at weblogic.security.acl.Realm.getRealm(Realm.java:62)
at weblogic.security.SecurityService.initializeRealm
(SecurityService.java:258)
at weblogic.security.SecurityService.initialize
(SecurityService.java:115)
at weblogic.t3.srvr.T3Srvr.initialize(T3Srvr.java:402)
at weblogic.t3.srvr.T3Srvr.run(T3Srvr.java:202)
at weblogic.Server.main(Server.java:35)
'>

```

**Solution**—Ensure the user mapped to weblogic\_system user is the delegated administrator for the policy domain specified for creating WebLogic Roles.

Ensure that Access Management Service is enabled on the Access Server and AccessGate.

Ensure that the servers have been restarted after making the following change:

```

OBWebPass.CreatUserWorkFlowID= wfqs20020806T0907402920,
obcontainerId=workflowDefinitions, OU=OblIx,
OU=Company,DC=qalab-vduong,DC=oblix,DC=com
OBWebPass.CreatUserWorkFlowDomain=
OU=Company,DC=qalab-vduong,DC=oblix,DC=com:

```

**Problem**—I see this exception:

```

<Aug 8, 2002 6:01:32 PM PDT> <Emergency> <Server> <Unable to
initialize the server: 'Fatal initialization exception
    Throwable: java.lang.UnsatisfiedLinkError: no obaccess in
    java.library.path

```

```

java.lang.UnsatisfiedLinkError: no obaccess in java.library.path
at java.lang.ClassLoader.loadLibrary(ClassLoader.java:1312)
at java.lang.Runtime.loadLibrary0(Runtime.java:749)
at java.lang.System.loadLibrary(System.java:820)
at com.oblix.access.ObConfig.<clinit>(ObConfig.java:152)
at com.oblix.realm.NetPointBEARealm.<init>
(NetPointBEARealm.java:239)
at java.lang.Class.newInstance0(Native Method)
at java.lang.Class.newInstance(Class.java:237)
at weblogic.security.acl.Realm.getRealm(Realm.java:84)
at weblogic.security.acl.Realm.getRealm(Realm.java:62)
at weblogic.security.SecurityService.initializeRealm
(SecurityService.java:258)
at weblogic.security.SecurityService.initialize
(SecurityService.java:115)
at weblogic.t3.srvr.T3Srvr.initialize(T3Srvr.java:402)
at weblogic.t3.srvr.T3Srvr.run(T3Srvr.java:202)
at weblogic.Server.main(Server.java:35)
'>

```

**Solution**—Check to make sure that *install\_dir/NetPointBEARealm/oblix/lib* is in the proper environment variable:

- **NT**—PATH
- **Solaris**—LD\_LIBRARY\_PATH

**Problem**—When using NetPoint with the BEA WebLogic Server with WebPass in SSL mode, both programmatic and console methods for creating and deleting a user may fail. A stack trace will state "Error making SOAP request null".

**Solution**—This error indicates that a conflict exists between the Java Secure Socket Extensions implementation provided with NetPoint and the implementation provided with the BEA WebLogic server.

Force NetPoint's JSSE implementation to load first by prepending the *jsse.jar* file located in the NetPoint installation lib directory to the WebLogic classpath.

Examples:

**Unix:**

```

set NP_HOME=/opt/netpoint/NetPointBEARealm/oblix
set CLASSPATH=$NP_HOME/lib/jsse.jar:$CLASSPATH:<additional
NetPoint Jars>
export CLASSPATH

```

**Win32:**

```
set NP_HOME=c:\netpoint\NetPointBEARealm\oblix
set CLASSPATH=%NP_HOME%\lib\jsse.jar;%CLASSPATH%:<additional
NetPoint Jars>
```

**Problem**—I see the following `java.lang.IllegalAccessError`:  
“`java.lang.NullPointerException` while booting the WebLogic server after installing NetPointBEARealm.”

```
<Jan 11, 2003 5:52:04 PM PST> <Info> <Security> <Getting boot
password from user.>
Enter password to boot webLogic server:
Starting webLogic Server ....
<Jan 11, 2003 5:52:08 PM PST> <Notice> <Management> <Loading
configuration file .\config\mydomain\config.xml ...>
<Jan 11, 2003 5:52:12 PM PST> <Emergency> <Server> <Unable to
initialize the server: 'Fatal initialization exception
Throwable: java.lang.IllegalAccessError:
java.lang.NullPointerException
java.lang.IllegalAccessError: java.lang.NullPointerException
at weblogic.security.acl.Realm.getRealm(Realm.java:91)
at weblogic.security.acl.Realm.getRealm(Realm.java:62)
at weblogic.security.SecurityService.initializeRealm
(SecurityService.java:261)
at weblogic.security.SecurityService.initialize
(SecurityService.java:118)
at weblogic.t3.srvr.T3Srvr.initialize(T3Srvr.java:418)
at weblogic.t3.srvr.T3Srvr.run(T3Srvr.java:212)
at weblogic.Server.main(Server.java:35)
'>
```

The webLogic Server did not start up properly.

Exception raised:

```
java.lang.IllegalAccessError: java.lang.NullPointerException
at weblogic.security.acl.Realm.getRealm(Realm.java:91)
at weblogic.security.acl.Realm.getRealm(Realm.java:62)
at weblogic.security.SecurityService.initializeRealm
(SecurityService.java:261)
at weblogic.security.SecurityService.initialize
(SecurityService.java:118)
at weblogic.t3.srvr.T3Srvr.initialize(T3Srvr.java:418)
at weblogic.t3.srvr.T3Srvr.run(T3Srvr.java:212)
at weblogic.Server.main(Server.java:35)
Reason: Fatal initialization exception
```

**Solution**—This exception occurs when the NetPointBEARealm does not find its configuration file, `NetPointBEARealm.properties`. The `NetPointBEARealm.properties` file must be placed in the same directory as the configuration file of the WebLogic Server, `config.xml`.

**Problem**—There is a requirement to be able to implement different authentication schemes for different WebLogic server instances. For example, one WebLogic server is to use LDAP authentication and a second is to use NT authentication.

**Solution**—In the Access Manager, create an authentication resource called / Authen/NT (or another appropriate name) of type Authen. Protect this resource with a policy and define an appropriate authentication rule. Set an authorization rule for the policy that authorizes everyone and has the following actions:

**Type**—WL\_REALM

**Name**—uid

**User attribute**—*login attribute*

Configure NetPoint BEARealm to use this authentication scheme using the following new parameters:

OBAuthnScheme1.ResourceName=/Authen/NT

OBAuthnScheme1.OBUserCred=*credentials for authentication scheme*

OBAuthnScheme1.OBPasswdCred=*password for authentication scheme*

For example, the *credentials for authentication scheme* can be *ntuserid* and the *password for authentication scheme* can be *ntpassword*. A realm that is configured with these parameters will service all authentication requests using this authentication scheme.

The WebLogic server will run as the user mapped to the system user in the directory server. The WebLogic server will use Basic over LDAP authentication for only this user. The external NT authentication scheme does not need to have a definition for this system user. For example, a system user can be mapped to *weblogic\_system* in the directory, but this user would not be required to have an NT account.

**Problem**—Single sign-on does not work with Apache Reverse Proxy.

**Solution**—When configuring single sign-on with BEA WebLogic using a reverse proxy, for the first authentication request to WebGate running on Apache Reverse Proxy, the Proxy does not pass the ObSSOCookie to WebLogic after successful authentication. To avoid this issue, use Form Based authentication instead of Basic Over LDAP when using Apache Reverse Proxy with BEA WebLogic.

# 3 Integrating NetPoint Security Provider for WebLogic

This chapter describes how to use NetPoint 7.0 with BEA WebLogic running in a Security Service Provider Interface (SSPI) implementation. WebLogic provides an environment for creating, integrating, securing, and managing distributed Java applications. The NetPoint 7.0 Security Provider for WebLogic ensures that only appropriate users and groups can access NetPoint-protected WebLogic resources to perform specific operations. The NetPoint Security Provider also allows you to configure single sign-on between NetPoint and WebLogic resources.

---

**Note:** The NetPoint Ready Realm for BEA supports WebLogic running in compatibility mode. See “Integrating NetPoint Ready Realm for BEA” on page 27 for details.

---

This chapter covers the following topics:

- “About the NetPoint Security Provider” on page 102
- “Integration Architecture” on page 104
- “Supported Versions and Platforms” on page 110
- “Online Assistance” on page 111
- “Installing and Configuring the Security Provider” on page 112
- “Configuring SSO for the Portal Server” on page 141
- “Configuration Files” on page 151
- “Implementation Notes for Active Directory” on page 161
- “Tips” on page 165
- “References” on page 169
- “Troubleshooting NetPoint Security Provider for WebLogic” on page 170
- “Additional Resources” on page 176

# About the NetPoint Security Provider

The NetPoint Security Provider for WebLogic (the Security Provider) provides authentication, authorization, and single sign-on across J2EE applications that are deployed in the BEA WebLogic platform. The Security Provider enables WebLogic administrators to use NetPoint to control user access to business applications.

---

**Note:** The COREid integration with WebLogic supports only one policy domain. All WebLogic policies must reside in this single policy domain.

---

The Security Provider provides authentication to BEA WebLogic Portal resources and supports single sign-on between NetPoint and BEA WebLogic Portal Web applications. Apart from this, the security provider also offers user and group management functions.

## WebLogic and NetPoint Integration Points

The WebLogic security framework provides Security Service Provider Interfaces (SSPIs) to protect J2EE applications. The NetPoint Security Provider for WebLogic takes advantage of these SSPIs, enabling you to use NetPoint to protect WebLogic resources via:

- User authentication
- User authorization
- Role mapping

The NetPoint Security Provider consists of several individual security providers. Each security provider enables a specific NetPoint function for WebLogic users:

**NetPoint Authenticator**—This security provider uses NetPoint authentication services to authenticate users who access WebLogic applications. Users are authenticated based on their credentials, such as user name and password.

This security provider also offers user and group management functions. It enables the creation and deletion of users and groups from the BEA WebLogic Portal Admin tools. It also provides single sign-on between WebGates and portals.

**NetPoint Identity Asserter**—Like the NetPoint Authenticator, this security provider uses NetPoint's authentication services to validate already-authenticated NetPoint users using the ObSSOCookie and to create a WebLogic-authenticated session.

Whether you use the Authenticator or the Identity Asserter depends on your deployment scenario. See “Integration Architecture” on page 104 for details.

**NetPoint Authorizer**—This security provider uses NetPoint’s authorization services to authorize users who are accessing a NetPoint-protected resource. The authorization is based on NetPoint policies.

**NetPoint Role Mapper**—This security provider returns security roles for a user. These roles are defined in NetPoint, and they are provided by NetPoint using return actions on a special authentication policy. This authentication policy contains a resource with a URL prefix of /Authen/Roles. These roles are mapped to security roles in WebLogic.

**NetPoint Deployment Provider**—This security provider monitors the applications that are deployed or undeployed on the WebLogic Server and writes information about these applications to either NetPointDeployPolicy.txt or NetPointUndeployPolicy.txt. A special authorization rule for administrators in NetPoint provides access to the WebLogic applications described in NetPointDeployPolicy.txt.

## Differences From NetPoint Ready Realm for BEA

The NetPoint Security Provider for WebLogic is different from the NetPoint Ready Realm for BEA:

- Ready Realm works only in WebLogic’s compatibility mode, which uses an older security architecture.
- In Ready Realm, IPValidation parameter in the WebGateStatic.lst had to be set to false. This is not needed with the NetPoint Security Provider.
- With the NetPoint Security Provider, you can take advantage of a NetPoint challenge parameter that ensures the ObSSOCookie is only sent over an SSL connection and prevents the cookie from being sent back to a non-secure Web server.
- If you have the Ready Realm roles J2EE\_ROLE or J2EE\_ACL defined, the new SSPI implementation does not support them.
- There is no upgrade path from the NetPoint Ready Realm for BEA to the NetPoint Security Provider for WebLogic.

# Integration Architecture

With the NetPoint Security Provider, you can use NetPoint to protect WebLogic resources, including Web applications, EJBs, JNDIs, and so on. You can configure single sign-on for Web applications, such that a user who has authenticated to WebLogic can access NetPoint-protected resources (including WebLogic and non-WebLogic resources) without re-authentication. You can also configure single sign-on such that a user who has authenticated to NetPoint can access WebLogic resources without re-authentication.

---

**Note:** With the WebLogic Portal Server, the NetPoint Security Provider is used *only* to provide authentication to WebLogic resources. Role Mapping and Authorization is taken care by the Portal Server.

---

The following applies *only* to the WebLogic Application Server. For Portal Server authentication, see “Authentication for the Portal” on page 108.

You can use the NetPoint integration with WebLogic in the following ways:

- To provide authentication for mixed Web and non-Web resources.

This scenario assumes that the environment configuration does not use a proxy server running a WebGate. In this scenario, you protect resources using username and password authentication. This method can be used to protect both HTTP resources on the Web and to protect resources such as EJBs, JNDIs, and other types of applications. In this type of scenario, all J2EE Web application deployment descriptors must be configured to be BASIC or FORM. This method requires use of the NetPoint Authenticator service provider.

See “Authentication for Mixed Web and Non-Web Resources” on page 105 for details.

- To provide authentication for Web resources only using a proxy server with WebGate installed.

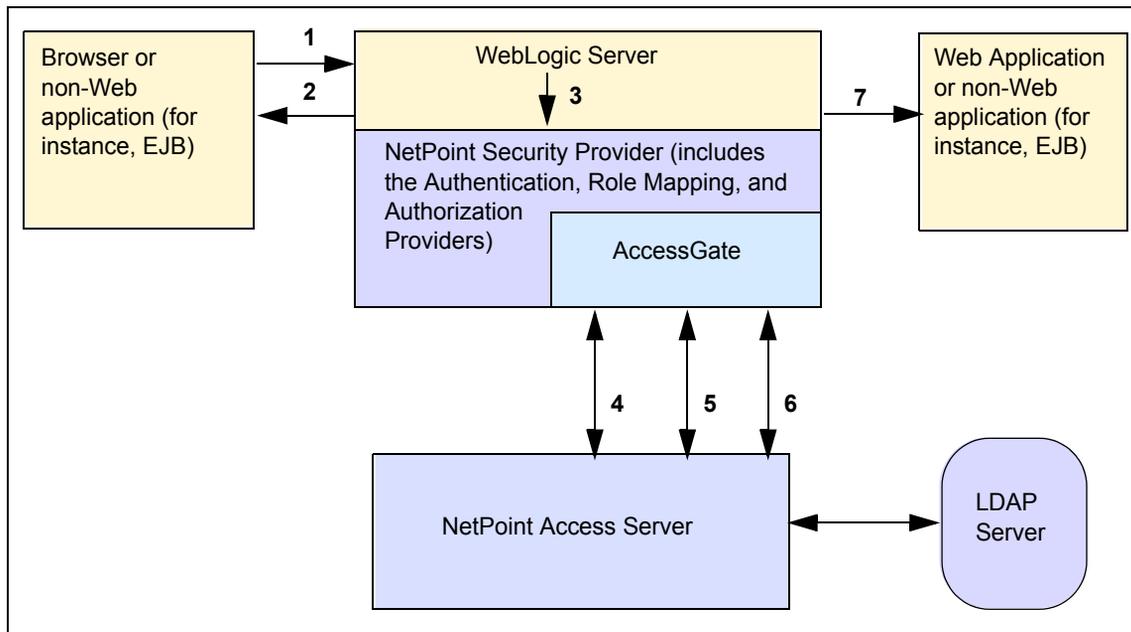
In this scenario, you protect resources using a variety of authentication schemes, such as form, SecurID, and so on. For this type of authentication to work, all J2EE Web applications must have an authen-method deployment descriptor configured to CLIENT-CERT. This configuration uses the NetPoint Identity Assertion security provider. See “Authentication for Web-Only Resources” on page 106 for details.

Non-Web applications must use the NetPoint Authenticator service provider rather than the NetPoint Identity Assertion security provider.

## Authentication for Mixed Web and Non-Web Resources

In this scenario, WebLogic resources, including both Web and non-Web applications, are protected using NetPoint authentication and authorization schemes. In this scenario, the J2EE Web applications have been configured with deployment descriptors for basic or form authentication. There is no WebGate protecting the WebLogic Server. AccessGate generates the ObSSOCookie when the user is authenticated, and AccessGate handles all the communications between the NetPoint Security Provider and the NetPoint Access Server.

**Figure 2** Mixed Web and non-Web Resources (Basic and Form Authentication)



### Process overview: User authentication, mixed resource types

1. A user attempts to access a NetPoint-protected WebLogic resource.
2. The WebLogic Server challenges the user for a username and password.
3. The WebLogic Server forwards the username and password to the NetPoint Security Provider for authentication and authorization.
4. The NetPoint Authentication Provider uses the AccessGate to communicate with the NetPoint Access Server to verify the user's identity.
5. If authentication is successful, the NetPoint Role Mapping Provider uses the AccessGate to communicate with the NetPoint Access Server to determine what NetPoint-defined roles are assigned to this user. These roles are mapped to security roles in WebLogic. In NetPoint, these roles are configured as a

return action when getting an authorization policy for /Authen/Roles. The return actions can be configured in three ways:

- **Static**—By entering constant values for name and return value.
  - **Dynamic**—By configuring a user profile attribute as the return value.
  - **Dynamic**—By configuring a special attribute called obmygroups and its derivatives as the return attribute. This returns all the groups that the user belongs to.
6. The NetPoint Authorization Provider uses the AccessGate to ask the NetPoint Access Server to verify that the user has permission to access the requested resource. This provider supports context sensitive authorization. See “Context-Specific Authorization” on page 146 for details.
  7. If authorization is successful, the WebLogic Server allows the user to access the requested resource. The ObSSOCookie is set so that when the user attempts to access additional NetPoint-protected non-WebLogic resources, re authentication is not performed.

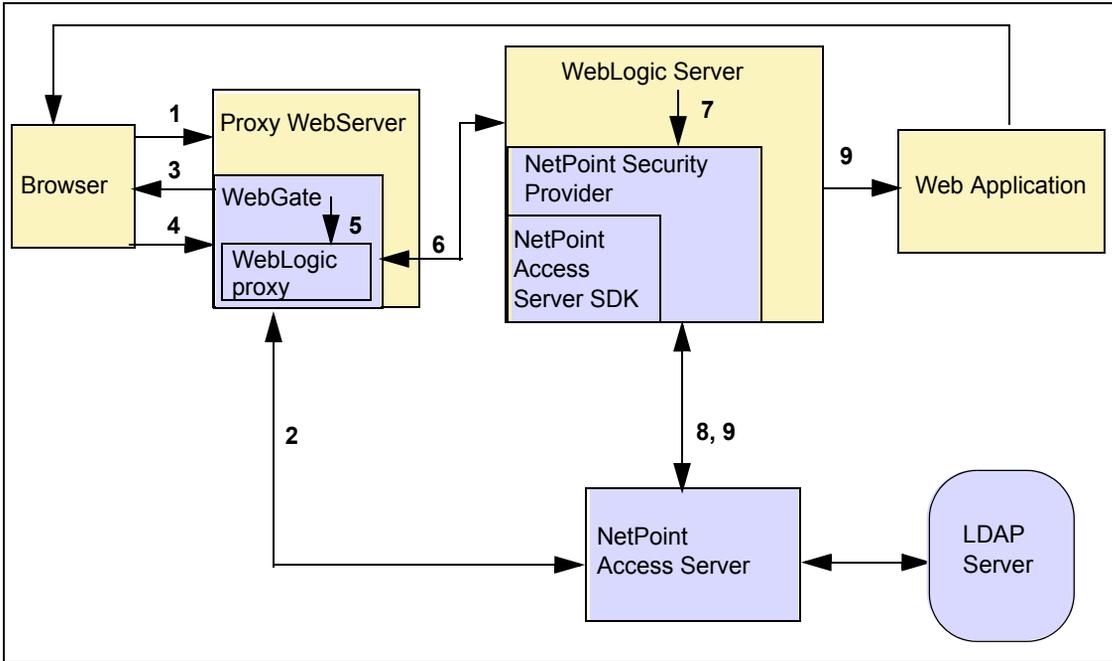
In this scenario, if the ObSSOCookie is already set and the user has logged in using form-based authentication, the user is logged in without being challenged. You can configure this type of integration by providing additional logic as illustrated in the sample file WebLogic login.jsp. You provide the additional logic in a file called NetPointSSO.jsp, which is provided in the installation directory.

## Authentication for Web-Only Resources

In this scenario, all types of authentication schemes supported by NetPoint can be used, including those that require identity assertion (also called perimeter authentication), using the ObSSOCookie as the basis of the authentication. A proxy server running WebGate is installed to protect the WebLogic Server. The WebGate performs all of the authentications and authorizations. Identity assertion is used for authenticating Web applications in WebLogic.

This scenario only supports Web applications.

**Figure 3** NetPoint Security Provider Scenario for Client Cert Authentication



**Process overview: User authentication, Web-only applications**

1. A user attempts to access a NetPoint-protected Web application that is deployed on the WebLogic server.  
The application has an authen-method deployment descriptor configured to CLIENT-CERT.
2. WebGate intercepts the request and queries the Access Server to check if the resource is protected.
3. If the resource is protected, WebGate challenges the user for credentials based on the type of NetPoint authentication scheme configured for the resource.
4. The user presents credentials such as user name and password, or a certificate.
5. If user authenticates successfully, WebGate generates an ObSSOCookie, sets it in the HTTP header, and passes it to a WebLogic proxy plug-in on the WebGate Web server.
6. The WebLogic proxy plug-in passes the cookie to the WebLogic Server.
7. The WebLogic Server's security service invokes the NetPoint Identity Assertion Provider.
8. The NetPoint Identity Assertion Provider extracts the ObSSOCookie information from the HTTP header, validates the ObSSOCookie, and retrieves

the user identity from the Access Server using a return action defined in a special authentication scheme in NetPoint. This authentication scheme contains a resource with a URL prefix of /Authen/Basic.

9. The remaining steps are the same as for “Process overview: User authentication, mixed resource types” on page 105, step 5 - step 7.

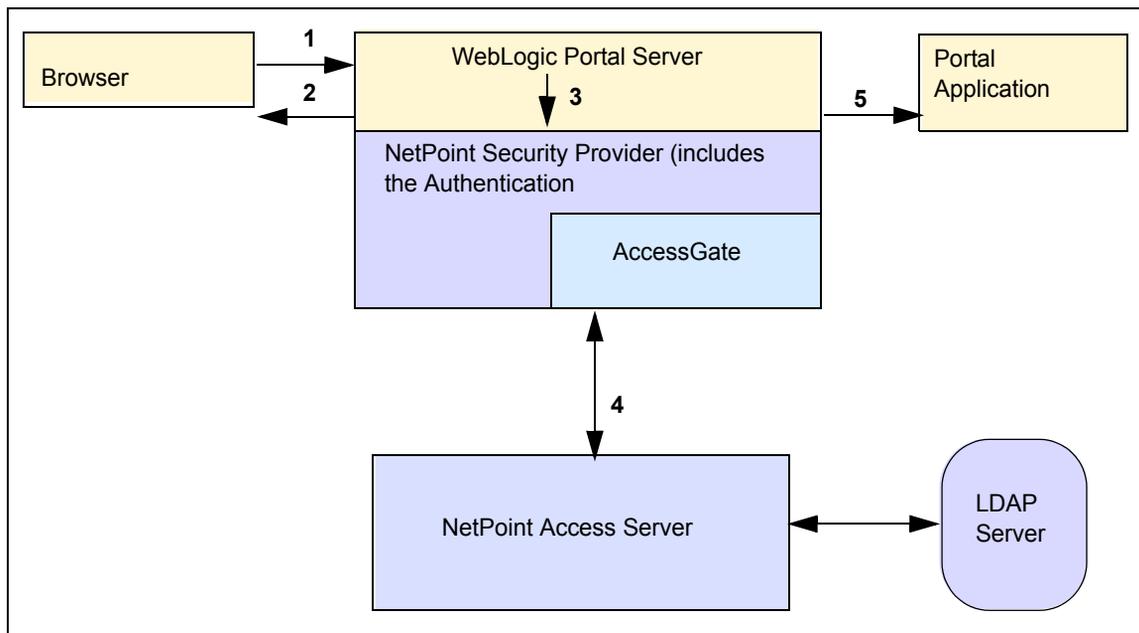
## Authentication for the Portal

NetPoint 7.0.3 and later version support integration with the WebLogic Portal Server.

The following scenario assumes that the environment configuration does *not* use a proxy Server running a WebGate. In this scenario, the resource is *not* required to be a NetPoint protected resource.

The connector internally authenticates the resource against the configured resource /Authen/Basic. The Portal application needs to be an application protected by WebLogic with username and password authentication. See “User authentication for the Portal” on page 108.

**Figure 4** User authentication for the Portal



### Process overview: User authentication for the Portal

1. A user attempts to access a protected WebLogic resource.
2. The WebLogic Server challenges the user for username and password.

3. The WebLogic Server forwards the username and password to the NetPoint Security Provider for authentication.
4. The NetPoint Authentication Provider uses the AccessGate to communicate with the NetPoint Access Server to verify the user's identity.  
  
If authentication is successful, the NetPoint Authentication Provider sets the subject correctly and passes control to WebLogic for Role Mapping and Authorization.
5. WebLogic displays the Portal application, on the basis of the authorization granted to the various portlets, etc. in the Portal Application.

**SSO**—Single sign-on between NetPoint-protected non-WebLogic resources to WebLogic Resources and vice-versa can be achieved for Portal Web Applications that are authenticated using a login portlet.

To achieve this, additional logic must be added in form of `NetpointPortalSSO.jsp` to the `login.jsp` used by the login portlet and the POST action of the login form must be configured to invoke the Oblix login filter class. For SSO setup details are given in section, see “Configuring SSO for the Portal Server” on page 141.

### **Process overview: SSO between NetPoint-protected non-WebLogic resources to WebLogic resources**

1. A user accesses NetPoint-protected non-WebLogic resources and the `ObSSOCookie` is set.
2. A user accesses a WebLogic resource.
3. The `NetpointPortalSSO.jsp`, which is included as a part of `login.jsp`, intercepts the `ObSSOCookie` and authenticates using the `ObSSOCookie`.  
  
In this case login form present in `login.jsp` is *not* displayed.
4. WebLogic authorizes the resources in the Portal Application.

### **Process overview: SSO between WebLogic resources to NetPoint-protected non-WebLogic resources**

1. A user tries to access a WebLogic resource.
2. The `NetpointPortalSSO.jsp`, which is included as a part of `login.jsp`, checks for the `ObSSOCookie`.  
  
In this case, no cookie is set and the login form present in `login.jsp` is displayed. Authentication occurs using the `ObSSOCookie`.
3. The user enters their credentials in the login form; the credentials get posted to the Oblix Login Filter (`ObLoginFilter` configured in `web.xml`).

4. The Oblix Login Filter authenticates the user with Access SDK; if authentication is successful, the Oblix Login Filter sets the ObSSOCookie and redirects to the main resource.

Authentication is *not* done using WebLogic authentication.

5. The control reaches to login.jsp; with the ObSSOCookie set, flow mentioned above is followed and WebLogic Portal is accessed. See “Process overview: SSO between NetPoint-protected non-WebLogic resources to WebLogic resources” on page 109.
6. The user accesses NetPoint-protected non-WebLogic resources with the ObSSOCookie set and no re-authentication performed.

## Supported Versions and Platforms

The following table identifies NetPoint 7.0.3 support for the WebLogic Portal Server.

**Table 4** NetPoint 7.0.3 Support

Operating System	WebLogic Application and Portal Server
Red Hat Enterprise Linux AS 3.0	v8.1 SP3 (via SSPI)
Windows 2000 Advanced Server (SP4)	v8.1 SP3 (via SSPI)
Windows 2003 Standard Server	v8.1 SP3 (via SSPI)

---

**Note:** BEA WebLogic 7.0 and WebLogic Portal 7.0 are supported via compatibility. WebLogic 6.1 and WebLogic Portal 4.0 are deprecated and support is not planned for future releases of NetPoint.

---

Table 5 shows NetPoint 7.0 support for the WebLogic Application Server:

**Table 5** NetPoint 7.0 Support

Operating System	WebLogic Server
Solaris 8	v7.0 SP 2 and higher v8.1
HP-UX 11.0	v8.1
Red Hat Enterprise Linux AS 2.1	v8.1
Windows 2000 Advanced Server (SP4)	v7.0 SP 2 and higher v8.1
Windows 2003 Standard Server	v7.0 SP 2 and higher v 8.1

## Online Assistance

Information about installing and configuring the components required for integration of NetPoint and WebLogic is provided in the following sections of this chapter and in a readme file. For access to the readme, go to:

*NetPoint\_Security\_Provider\_install\_dir*

where *NetPoint\_Security\_Provider\_install\_dir* is the directory where the NetPoint Security Provider was installed.

# Installing and Configuring the Security Provider

The following sections provide the information needed to install and configure the NetPoint Security Provider for WebLogic.

## Preparing the Environment

Before you install NetPoint Security Provider for WebLogic, complete the following tasks:

### **Task overview: Before installing the NetPoint Security Provider for WebLogic**

1. Install and set up the WebLogic Application/Portal Server as described in your vendor documentation.

---

**Note:** NetPoint 7.0.3 supports integration with the WebLogic Portal Server.

---

2. Be sure you are using JDK 1.4.
3. Install and set up NetPoint COREid System and Access System, as described in the *NetPoint 7.0 Installation Guide*:
  - In the NetPoint Access Manager, add an AccessGate and associate it with the Access Server that you installed.

Add and associate an AccessGate with an Access Server for the NetPoint Security Provider for WebLogic. You may want to name the AccessGate accordingly; for example, *WebLogicProvider*.

---

**Note:** If you are going to use the deployment tool discussed in “Running the NetPoint Policy Deployer” on page 122, you must turn on the Access Management Service for the AccessGate, as well as all Access Servers associated with the AccessGate.

---

4. Create a user in NetPoint who is a WebLogic administrator and give this person delegated administrative rights. See the chapter on policy domains in the *NetPoint 7.0 Administration Guide Volume 2* for details.
5. Install NetPoint Security Provider for WebLogic as described in “Installing NetPoint Security Provider for WebLogic” on page 113.

## Installing NetPoint Security Provider for WebLogic

The installation procedure depends on the platform on which you are installing NetPoint. The example below occurs on a Windows system; however, installation is the same after you launch the installation package for your platform.

### To install the NetPoint Security Provider for WebLogic

1. Locate and launch the NetPoint Security Provider for WebLogic installation package.
2. Execute `NetPoint7_0_3_Win32_BEA_WL_SSPI`.

The install wizard launches and the Welcome screen appears.

If the AccessGate fails during installation, you can run the tool `configureAccessGate` after installation, which is located in:

```
NetPoint_Security_Provider_install_dir\oblix\tools\configureaccessgate
```

where *NetPoint\_Security\_Provider\_install\_dir* is the directory where the Security Provider is installed. See the *NetPoint 7.0 Administration Guide Volume 2* for information on AccessGates and the `configureAccessGate` tool.

3. On the Welcome screen, click Next.  
The License screen appears.
4. Read the License screen, accept the license, then click Next.
5. Confirm that you are logged in as a user with administrative rights, then click Next.
6. Select an installation directory and click Next.
7. View the confirmation screen and click Next.  
A set of files are installed. When the installer has completed, you are prompted as to whether you want a Typical or Advanced installation.
8. Select Typical or Advanced and click Next.
9. Continue with the procedure that is appropriate for your environment:
  - “Completing a Typical Installation” on page 114
  - “Completing Advanced Installation” on page 115

## Completing a Typical Installation

Typical installations supply default values for the Security Provider configuration.

A typical installation prompts you for a transport security mode. The transport security mode that you select for the Security Provider must match the transport security mode for the Access Server. Information on the prompts for installing in simple and certificate mode are available in the *NetPoint 7.0 Installation Guide*. The prompts for configuring the transport security mode for the NetPoint Security Provider are similar to those presented when installing any other NetPoint component.

### To finish a typical installation

1. Complete WebPass details, as follows:

- a) Enter the hostname where webpass is installed
- b) Enter the webpass port number
- c) Indicate whether the webpass is protected by a webgate

You complete step 2 when the WebPass is protected by a WebGate. Otherwise, proceed to step 3. If you have chosen to use WebGate to protect WebPass, the assumption is that you are protecting the NetPoint applications with policy domains. Therefore, it is also assumed that single sign-on between these components has been configured correctly.

2. **WebPass Protected by WebGate**—Complete the steps below.

- a) Enter the cookie domain for the WebGate (for example, .domain.com). The ObSSOCookie is then recognized by all servers within this domain.
- b) Enter the cookie path (/).

3. Complete directory-specific information, as shown below:

- a) Specify whether WebPass requires an HTTPS connection.

This is the SSL for secure connection when WebPass runs on HTTPS.

- b) Specify the user attribute.

This attribute must be the same as the attribute configured for the Login semantic type in the COREid Server or a unique attribute in the user's profile such as uid.

- c) Specify the user search attribute.

This attribute must be the same as the attribute configured for the DN Prefix semantic type for the person object class in the COREid Server. The person object class type must be a structural object class. The administrator of your directory server sets this search attribute. The user attribute and the user search attribute *cannot* be the same.

- d) Specify the group search attribute.

This attribute must be the same as the attribute configured for the DN Prefix semantic type for the group object class in the COREid Server. The group object class a structural object class. The administrator of your directory server sets the group search attribute.

4. Select a transport security mode:
  - **Open**—If you select open mode, all data is in plain text.
  - **Simple**—If you select simple mode, you are prompted to supply a global pass phrase. As in Cert mode, you secure the private key with a Privacy Enhanced Mail (PEM) pass phrase. Before an AccessGate or Access Server can use a private key, it must have the correct PEM pass phrase. The PEM pass phrase is stored in a encrypted file called password.lst. For Simple mode, the PEM pass phrase is the same for each WebGate and Access Server instance.
  - **Cert**—If you select cert mode, you are prompted to supply a global pass phrase. You are then asked if you wish to request a certificate or install a certificate.
5. Supply information regarding the AccessGate and Access Server that you have installed.
6. Review the readme that appears.

The information in this readme is covered in this chapter also.
7. Confirm the installation.
8. Continue with “Setting Up WebLogic Policies in NetPoint” on page 117.

## Completing Advanced Installation

Advanced installation permits you to override the default configuration. All the configuration options that you can set in an advanced installation are provided in the sample configuration file described in “NetPointProvidersConfig.properties” on page 152. This gives you the opportunity to customize your installation, which can be useful if you have configured several versions of the Security Provider authentication and authorization schemes.

---

**Note:** Do *not* attempt an advanced installation unless you are familiar with creating policy domains and policies in NetPoint and have run through at least one typical installation of the NetPoint Security Provider.

---

## To finish an advanced installation

1. Complete the screen subtitled “NetPoint Security Provider” uses a special policy to authenticate users in WebLogic. Please specify the following:
  - **Resource Type**—This is the name of a resource type used by the Security Provider to authenticate users. See “To configure the WebLogic resource types” on page 118 for details.
  - **Resource Name**—This is the URL prefix for the resource used by the Security Provider to authenticate users. See “To add resources to the domain in NetPoint” on page 125 for details.
  - **Resource Name used for Anonymous Access**—This is the URL prefix for the resource used when allowing anonymous access to certain resources. See “To add resources to the domain in NetPoint” on page 125 for details.
  - **Resource operation**—This is the operation specified on the resource type definition. The operation is performed to authenticate users.
  - **Login Parameter for credential\_mapping Plug-in of Authentication Scheme**—See “To create WebLogic authentication schemes” on page 119 for details.
  - **Password Parameter User for password\_validation Plug-in of Authentication Scheme**—See “To create WebLogic authentication schemes” on page 119 for details.
  - **Action Type**—Action is configured to get the loginID from the ObSSOCookie. This is the action configured on the authorization rule. See “To add authorization and authentication rules to the domain” on page 126 for details.
  - **Action Name**—Action is configured to get the loginID from the ObSSOCookie. See “To add authorization and authentication rules to the domain” on page 126 for details.
  - **Dummy Username**—For Form Login with SSO when there is No WebGate on Proxy HTTP Server. This is used if you are protecting both Web and non-Web resources and you are using form login. If the login.jsp is modified to include NetPointSSO.jsp, and the user has already logged in to a NetPoint-protected resource, the user has received a token. The next time the user tries to access a protected resource, NetPoint uses the dummy user name as the user name and the token is used as the password. The default is obdummyuser. Oblix recommends that you use the default name.
  - **WebLogic resource types used for web applications (comma separated)**—These are the resource types that WebLogic uses for Web applications. See “Mapping WebLogic Resources to NetPoint Resources” on page 131 for details.

2. Complete the screen subtitled “NetPoint Security Provider uses a special policy to get roles for a user. Please specify the following configuration to set up this policy”:
  - **TTL (time to live) of elements in roles cache**—This is the amount of time the action is preserved in the cache.
  - **Time to delete expired elements of cache (in seconds)**—This is the time interval for freeing the memory used for expired elements in the cache.
  - **Resource type**—This is the name of a resource type used by the Security Provider to get roles. See “To configure the WebLogic resource types” on page 118 for details.
  - **Resource name**—This is the URL prefix for the resource used by the Security Provider to get roles. See “To add resources to the domain in NetPoint” on page 125 for details.
  - **Resource operation**—This is the operation specified on the resource type definition. The operation is performed to authenticate users.
  - **Action Type in authorization rule to get roles**—This is the action configured on the authorization rule to get user roles. See “To add authorization and authentication rules to the domain” on page 126 for details.
3. Complete the first screen, “Configuration for NetPoint Security Provider for WebLogic”:
  - **Default access to resources not protected by NetPoint (deny, allow, abstain)**—Allow grants access, deny forbids it. Abstain means that if there are multiple security providers for WebLogic, WebLogic goes to the next security provider to decide what to do.
  - **Map the authorization result ABSTAIN to (allow, deny)**—A result of abstain can be automatically reset to allow or deny.
  - **Set debugging**—Debug logs are written to the WebLogic log file.
4. Finish the installation by completing the steps described in “Completing a Typical Installation” on page 114.

## Setting Up WebLogic Policies in NetPoint

After installing NetPoint and WebLogic, you need to define NetPoint policy domains that provide a method for protecting WebLogic applications. The basics of defining a policy domain are:

- **Creating resource types**—This allows NetPoint to identify the kinds of WebLogic resources that it should protect and the operations (such as GET) associated with the resource.

- **Creating authentication schemes**—This allows NetPoint to verify user identities.
- **Creating authorization schemes**—This allows NetPoint to grant users access to the resources that you have defined.
- **Creating a policy domain**—This creates a container for your WebLogic-related policies.
- **Creating policies**—These are directives for protecting specific WebLogic resources. Policies are an amalgam of resource type definitions, URLs identifying the resource locations, and the authentication and authorization schemes to apply when users access the resources.

The first step in setting up a policy domain is to define your resources and authentication schemes. These tasks are discussed in the following sections, which assume a basic knowledge of NetPoint. The *NetPoint 7.0 Administration Guide Volume 2* provides details on tasks described in the following sections.

---

**Note:** The resource type `wl_svr` described below is available if you want to protect access to starting and stopping the WebLogic server. To do this, you can define a policy (as described in “To create policies for the domain” on page 129) that uses this resource type. You can find the information you need for this policy in the `isAccessAllowed` entries in the debug logs that contain the string `<svr>`. Note that you must create this policy manually. The deployment tool provided with the NetPoint Security Provider does not create this policy for you. Also, the resource type `wl_ejb` is used when you deploy EJB applications. You can create policies that use this type of resource manually, or you can use the deployer tool to create these policies.

---

### To configure the WebLogic resource types

1. From the Access System Console, click the Access System Configuration tab, then click Common Information Configuration > Resource Type Definitions.

The List All Resource Types page appears.

2. From the List All Resource Types page, click Add.

The Define a new Resource Type page appears.

3. Define and save the first resource type:

**Name**—`wl_url`

**Display name**—`wl_url`

**Resource matching**—case insensitive

**Resource operation**—GET, POST

4. Define and save the second resource type:

**Name**—wl\_svr  
**Display name**—wl\_svr  
**Resource matching**—case insensitive  
**Resource operation**—BOOT, DEFAULT

5. Define and save the third resource type:

**Third resource:**  
**Name**—wl\_adm  
**Display name**—wl\_adm  
**Resource matching**—case insensitive  
**Resource operation**—DEFAULT

6. Define and save the fourth resource type:

**Name**—wl\_ejb  
**Display name**—wl\_ejb  
**Resource matching**—case insensitive  
**Resource operation**—EXECUTE

7. Define and save the fifth resource type:

**Name**—wl\_authen  
**Display name**—wl\_authen  
**Resource matching**—case insensitive  
**Resource operation**—LOGIN

8. Define and save the sixth resource type:

**Name**—http  
**Display name**—http  
**Resource matching**—case insensitive  
**Resource operation**—GET, POST, PUT, HEAD, DELETE, TRACE, OPTIONS, CONNECT, OTHER

## To create WebLogic authentication schemes

1. From the Access System Console, click Access System Configuration > Authentication Management > Add
2. Create the first authentication scheme.

This scheme is used by the Security Provider to authenticate users. Refer to the authentication scheme NetPoint Basic Over LDAP as a template:

- a) Configure the General tab:  
**Name**—NetPoint WebLogic Basic Over LDAP  
**Description**—Used to authenticate users who access WebLogic resources.  
**Level**—1  
**Challenge Method**—Basic

**Challenge Parameter**—realm:NetPoint Basic Over LDAP  
**SSL Required**—No  
**Challenge Redirect**—(Leave blank)  
**Enabled**—(Leave as is)

---

**Note:** The realm: string on the challenge parameter is required. The text after this parameter is configurable. Also, in the Name field, the “l” in WebLogic must be lowercase to match the name in a NetPointWeblogicTools.properties file that is part of the integration solution. In general, the name of this authentication scheme should be identical (with case sensitivity) to the ObWLAAuthenticationScheme.Name parameter in the NetPointWeblogicTools.properties file.

---

- b) Save the information on the General tab by clicking Save.
- c) Click the Plugins tab, and use the credential\_mapping and validate\_password plugins from the existing NetPoint authentication schemes. In the credential\_mapping plug-in, be sure the mapping base and mapping filter use objects that are specific to your environment. Examples:

```
validate_password obCredentialPassword="password"  
credential_mapping  
obMappingBase="o=company,c=us",obMappingFilter=  
"(&(&(objectclass=inetorgperson)(uid=%userid%))  
(!(obuseraccountcontrol=*))  
(obuseraccountcontrol=ACTIVATED)))"
```

where place-holders such as *o=company,c=us*, and *inetorgperson* are replaced with values that are valid for your organization.

After you create at least one plug-in, default steps and a default authentication flow are created automatically.

- d) After creating a plug-in, you can enable the authentication scheme by clicking the General tab > Modify, selecting the Enable option, and clicking Save.

3. Create the second authentication scheme, using NetPoint None Authentication as a template.

This scheme is used for un-protecting certain resources, such as gif images in WebLogic resources.

Follow the process for creating an authentication scheme shown in the previous step, using the following data:

**General tab:**

**Name**—NetPoint WebLogic Anonymous Authentication

**Description**—Used to un-protect gifs, and so on.

**Level**—0

**Challenge Method**—None

**Challenge Parameter**—(Leave blank)

**SSL Required**—No

**Challenge Redirect**—(Leave blank)

**Enabled**—Yes

---

**Note:** In the Name field, the “P” in WebLogic must be lowercase to match the name in a NetPointWeblogicTools.properties file that is part of the integration solution. In general, the name of this authentication scheme should be identical (including case sensitivity) to the ObWLNoneAuthenticationScheme.Name parameter in the NetPointWeblogicTools.properties file.

---

**Plugins tab:**

Use the credential\_mapping plug-in from the pre-configured NetPoint None authentication schemes. In the credential\_mapping plug-in, be sure the mapping base and mapping filter use objects that are specific to your environment. Use OblixAnonymous as the mapping filter. Example:

```
credential_mapping
obMappingBase="o=company,c=us",obMappingFilter="
(uid=OblixAnonymous)"
```

where place-holders such as *o=company,c=us*, and *uid* should be replaced with values appropriate for your environment.

4. Restart the Access Server.

You are now ready to create the policies below:

- **Basic Authentication Policy**—This policy is used internally to authenticate users by evaluating the user name and login. The policy protects resources with a URL prefix of /Authen/Basic.
- **Role-based Authentication Policy**—This policy gets user roles. This policy protects resources with a URL prefix of /Authen/Role.

- **Anonymous Authentication Policy**—This policy provides anonymous access to gifs, and other resources. This policy protects resources with a URL prefix of /Authen/Anonymous.
- **Anonymous Authentication Policy (second)**—This policy allows anonymous access for users. This policy protects resources with a URL prefix of /Authen/Anonymous.
- **WebLogic Administrator Policy**—This policy allows access to the WebLogic administration console.

The NetPoint Policy Deployer Tool can automate this process, or you can create these policies manually.

5. Continue with one of the discussions below:
  - “Running the NetPoint Policy Deployer” on page 122
  - “Manually Configuring WebLogic Policies in NetPoint” on page 125

## Running the NetPoint Policy Deployer

After you have created your resource types and authentication schemes, you can run the NetPoint Policy Deployer for WebLogic tool. This tool enables you to:

- Create the NetPoint policy domain and policies during initial setup of the NetPoint Security Provider for WebLogic.
 

This policy domain uses the resource type `wl_authen` created in “To configure the WebLogic resource types” on page 118.
- Create and delete NetPoint policy domains and policies that protect WebLogic applications.

You need to run this tool at least once for initial setup. Afterwards, you can either manually create policies for applications deployed in WebLogic, or you can run this tool to automatically create them. Refer to the appropriate procedures below:

- “To prepare for running the NetPoint Policy Deployer Tool” on page 122
- “To run the NetPoint Policy Deployer Tool for the first time” on page 123
- “To run the NetPoint Policy Deployer Tool after the first time” on page 124

---

**Note:** If you do not want to use the NetPoint Deployer tool, you must manually configure WebLogic policies as described in “Manually Configuring WebLogic Policies in NetPoint” on page 125.

---

### To prepare for running the NetPoint Policy Deployer Tool

1. Add the following to CLASSPATH:
  - `NetPoint_Security_Provider_install_dir/oblix/tools/npWLTtools`

- *NetPoint\_Security\_Provider\_install\_dir/oblix/tools/npWLTools/npWLTools.jar*
- *NetPoint\_Security\_Provider\_install\_dir/oblix/lib/jobaccess.jar*

2. Add the following:

*NetPoint\_Security\_Provider\_install\_dir/oblix/lib*

On Windows, you add this to the PATH. On Solaris, you add this to LD\_LIBRARY\_PATH. On HP-UX, you add this to SHLIB\_PATH.

3. Ensure that the following configuration files are copied from *NetPoint\_Security\_Provider\_install\_dir* to the WebLogic domain directory:

- NetPointProvidersConfig.properties

See “NetPointProvidersConfig.properties” on page 152 for details.

- NetPointResourceMap.conf

See “Mapping WebLogic Resources to NetPoint Resources” on page 131 for details on configuring this file.

### **To run the NetPoint Policy Deployer Tool for the first time**

1. Review the following configuration file:

*NetPoint\_Security\_Provider\_install\_dir/oblix/tools/npWLTools/NetPointWeblogicTools.properties*

where *NetPoint\_Security\_Provider\_install\_dir* is the directory where the Security Provider is installed.

Be sure that `setupNetPointSSPIPolicies` is set to true (the default).

2. If you are running the WebLogic Web applications in Identity Assertion mode, configure the following parameters in the `NetPointWeblogicTools.properties` configuration file:

- `ObWLWebResource.usingIdentityAssertion`
- `ObWLWebResource.proxyPrefix`

See “NetPointWeblogicTools.properties” on page 160 for details.

3. From the command line, enter the following:

```
java com.oblix.weblogic.tools.NetPointPolicyDeployer userid
password
```

where *userid* and *password* belong to the login ID of the NetPoint administrator. For all the policies that this tool creates, it initially grants access to this userID only. Use JDK 1.4 to ensure that this command works as expected.

4. Go to the NetPoint Access Manager and check if the policies are created.  
See “To create a policy domain in NetPoint” on page 125 for details on the policy domains and how they should be configured.
5. After running the tool, go to the NetPoint Administration Console and provide the proper access to the policies that have been created.

---

**Note:** Setting up security policies in NetPoint is only required for protection of resources deployed on the BEA WebLogic Application Server. The connector only supports authentication for portals. If you are using the SSPI connector in the portal domain, then only the policies required for authentication need to be created. These are created when you run the policy deployer tool for the first time with the `ObWLTTools.SetupInitialNetPointSSPIPolicies` parameter set to true and the `ObWLTTools.DeployPolicy` and `ObWLTTools.UnDeployPolicy` parameters set to false in the `NetPointWeblogicTools.properties` file. Hence, for the portal domain, the section on "To run the NetPoint Policy Deployer Tool after the first time" can be ignored.

---

6. Ensure that the policy domain that was just created is enabled in NetPoint.
7. In the Access Manager, click My Policy Domains and click the WebLogic policy domain.  
The status of the policy domain should be enabled.
8. If the status of the policy domain is not enabled, click Modify and enable it.

### **To run the NetPoint Policy Deployer Tool after the first time**

1. Open the following configuration file:

```
NetPoint_Security_Provider_install_dir/oblix/tools/npWLTTools/  
NetPointWeblogicTools.properties
```

where *NetPoint\_Security\_Provider\_install\_dir* is the directory where the Security Provider is installed.

2. Be sure that `NetPointDeployPolicy.txt` and `NetPointUndeployPolicy.txt` are in the WebLogic domain directory.

These files are created the first time the deployer tool is run. The NetPoint Security Provider for WebLogic writes security policy data into this file when J2EE applications are deployed or undeployed. The NetPoint Policy Deployer reads this file to create policies.

3. Set the following:
  - `ObWLTTools.SetupInitialNetPointSSPIPolicies=false`
  - `ObWLTTools.DeployPolicy=true`
  - `ObWLTTools.UnDeployPolicy=true`

4. Follow step 2 - step 5 of “To run the NetPoint Policy Deployer Tool for the first time” on page 123.
5. Continue with “Mapping WebLogic Resources to NetPoint Resources” on page 131.

## Manually Configuring WebLogic Policies in NetPoint

If you do *not* wish to use the NetPoint Policy Deployer tool described in “Running the NetPoint Policy Deployer” on page 122, you can configure the policies manually in NetPoint.

### Task overview: Manually configuring WebLogic Policies in NetPoint

1. Create a WebLogic policy domain, as described in “To create a policy domain in NetPoint” on page 125.
2. Add resources to the domain, as described in “To add resources to the domain in NetPoint” on page 125.

Multiple resources can be defined for each resource type configured in “To configure the WebLogic resource types” on page 118. Resources provide URL prefixes under which various policies can be defined.

3. Add authorization and authentication rules to the domain, as described in “To add authorization and authentication rules to the domain” on page 126.
4. Define policies within the domain, as described in “To create policies for the domain” on page 129.

The URL prefix in the resource and the URL pattern in the policy together form the definition of the data to be protected by the policy.

### To create a policy domain in NetPoint

1. Log in to the Access System.
2. From the Access Manager, click Create Policy Domain.
3. Define the following policy domain:
  - Name**—SecuProvForWebLogic
  - Description**—Policy domain for WebLogic resources.
4. Save this policy domain.

### To add resources to the domain in NetPoint

1. From the Access Manager, click My Policy domains and select the new policy domain.
2. Click the Resources tab for the policy domain.

3. Click add, configure, and save the resource for user authentication:

**Resource type**—wl\_authen

**URL prefix**—/Authen/Basic

**Description**—Resource used by the Security Provider to authenticate users.

The resource type was defined in “To configure the WebLogic resource types” on page 118.

4. Click add, configure, and save the resource for anonymous access:

**Resource type**—wl\_authen

**URL prefix**—/Authen/Anonymous

**Description**—Resource used for anonymous authentication, where a session is created for anonymous users.

5. Click add, configure, and save the resource for returning user roles:

**Resource type**—wl\_authen

**URL prefix**—/Authen/Roles

**Description**—Resource used when the policy is configured to return roles that are mapped to security roles in WebLogic.

6. Click add, configure, and save the resource for protecting the WebLogic administration console:

**Resource type**—wl\_url

**URL prefix**—/console

**Description**—Optional. Protects the WebLogic administration console.

7. Click add, and save the resource for server administration:

**Resource type**—wl\_svr

**URL prefix**—/servername

**Description**—Optional. Restricts access for users who perform server administration tasks, such as starting and stopping the server.

---

**Note:** If you get an error, be sure that these resources are not already used by another policy domain.

---

## To add authorization and authentication rules to the domain

1. In the Access Manager, add the resources that you defined in “To configure the WebLogic resource types” on page 118 to this policy domain.

Access Manager > My Policy domains > *policy domain*

Next, you add authorization rules.

2. Click the Authorization Rules tab, click Add, and create the rule for administrators:

**General tab:**

**Name**—Authz rule for admin

**Description**—Authorization rule for an administrator. This rule provides administrator access to WebLogic applications.

**Enabled**—Yes

**Actions tab:**

Leave blank.

**Allow Access tab:**

**People**—Add users who are allowed to be WebLogic administrators.

3. Click Add, and create the rule for anonymous access:

**General tab:**

**Name**—Authz rule for anyone

**Description**—Allows anonymous access to resources

**Enabled**—Yes

**Actions tab:**

Leave blank.

**Allow Access tab:**

**Role**—Anyone

4. Click Add, and create the general rule for access:

**General tab:**

**Name**—Authz Rule for Authen

**Description**—Returns the user ID from the NetPoint ObSSOCookie using the return action configured on the Actions tab.

**Enabled**—Yes

**Actions tab:**

**Redirect to**—Leave blank

**On Authorization Success Return**—

Type—WL\_REALM

Name—uid

Return Attribute—*login ID attribute* where this is the attribute in your directory for the user login ID.

**On Authorization Failure Return**—Leave blank

**Allow Access tab:**

**Role**—Anyone.

5. Click Add, and create the rule for returning the user WebLogic role:

**General tab:**

**Name**—Authz rule for role

**Description**—Returns the user's role. These roles are hard-coded on the return actions, and they match administrative roles in WebLogic.

**Enabled**—Yes

**Actions tab:**

**Redirect to**—Leave blank

**On Authorization Success Return**—

Type—WL\_REALM

Name—role1

Return Value—Admin

Type—WL\_REALM

Name—role2

Return Value—Operator

Type—WL\_REALM

Name—role3

Return Value—Monitor

Type—WL\_REALM

Name—role4

Return Value—Deployer

**On Authorization Failure Return**—Leave blank

**Allow Access tab:**

**People**—People who are allowed to be the WebLogic administrator.

---

**Note:** The WebLogic administration console requires the administrator to have certain roles. These are hard-coded on the return actions of Authz rule for role, above. As an alternative, you can allow access to everyone rather than just the administrator, and control access by using a return attribute such as a user profile attribute or a special attribute called obMyGroups that returns all the groups that a user belongs to.

---

6. Create a default authorization rule that allows anonymous access, as follows:
  - a) Click the Default Rules tab > Authorization Rules > Add.
  - b) Select the rule for anonymous access that you created in the previous step.  
This is the rule called Authz rule for anyone.
  - c) Click Save.

If no policy is evaluated, the default rule provides anonymous access to everyone. This can be changed to meet the requirements of your environment.

7. Create a default rule that authenticates users for access to all resources that do not fall under a specific policy, as indicated below:

- a) Click the Default Rules tab > Authentication Rule > Add.

The General page appears.

- b) On this page, add the following default rule (or configure another one, if needed for your environment):

**Name**—The name for this rule is NetPoint WebLogic Anonymous Authentication.

**Authentication scheme**—Use the authentication scheme you created in “To create WebLogic authentication schemes” on page 119.

### To create policies for the domain

1. Create policies for this domain from the Access Manager:
2. Access System Console > Access Manager > *policy domain* > Policies tab.
3. Add the basic policy for this policy domain:

**General tab:**

**Name**—Basic authentication policy

**Description**—Authentication using basic LDAP username and password.

**Resource type**—wl\_authen

**Resource operation**—LOGIN

**Resource**—/Authen/Basic

**Authentication Rule tab:**

**Name**—Basic authentication rule

**Scheme**—Select the basic authentication scheme NetPoint WebLogic Basic Over LDAP that you created in “To create WebLogic authentication schemes” on page 119.

**Authorization Rule tab**—Add the rule Authz Rule for Authen that you created in “To add authorization and authentication rules to the domain” on page 126.

---

**Note:** If you are using identity assertion as the authentication mechanism that protects Web applications, see the notes in “Preparing the WebLogic Environment” on page 135 after configuring your authentication policies.

---

4. Add the anonymous access policy for this policy domain:

**Policy, General tab:**

**Name**—Anonymous authentication policy

**Description**—Authenticates anonymous users.

**Resource type**—wl\_authen

**Resource operation**—LOGIN

**Resource**—/Authen/Anonymous

**Authentication Rule tab:**

**Name**—Anonymous authentication rule

**Scheme**—Select the anonymous authentication scheme NetPoint WebLogic Anonymous Authentication that you created in “To create WebLogic authentication schemes” on page 119.

**Authorization Rule tab**—Add the rule Authz rule for anyone that you created in “To add authorization and authentication rules to the domain” on page 126.

5. Add the user role policy for this policy domain:

**General tab:**

**Name**—Role-based authentication policy

**Description**—Authenticates users and gets their WebLogic roles

**Resource type**—wl\_authen

**Resource operation**—LOGIN

**Resource**—/Authen/Roles

**Authentication Rule tab:**

**Name**—Role authentication rule

**Scheme**—Select the NetPoint WebLogic Basic over LDAP rule that you created in “To create WebLogic authentication schemes” on page 119.

**Authorization Rule tab**—Add the Authz rule for role that you created in “To add authorization and authentication rules to the domain” on page 126.

6. Add the “unprotect” policy for this policy domain:

**General tab:**

**Name**—Unprotect policy for gifs and other files

**Description**—Allow anonymous access to gif files

**Resource type**—wl\_url

**Resource operation**—GET,POST

**Resource**—all (if there are no resources defined, this defaults to all)

**URL pattern**—/.../\*.gif

**Authentication Rule tab:**

**Name**—WebLogic Domain None authentication rule

**Scheme**—Select the NetPoint WebLogic anonymous authentication rule that you created in “To create WebLogic authentication schemes” on page 119.

**Authorization Rule tab**—Add the Authz rule for anyone that you created in “To add authorization and authentication rules to the domain” on page 126.

7. Add the WebLogic administration console policy for this policy domain:

**General tab:**

**Name**—Policy for WebLogic admin console

**Description**—Allow administrator access to the WebLogic admin console

**Resource type**—wl\_url

**Resource operation**—GET

**Resource**—/console

**Authentication Rule tab:**

**Name**—WebLogic Domain default authentication rule

**Scheme**—Select the NetPoint WebLogic Basic over LDAP rule that you created in “To create WebLogic authentication schemes” on page 119.

**Authorization Rule tab**—Add the Authz rule for admin that you created in “To add authorization and authentication rules to the domain” on page 126.

For additional information, see the *NetPoint 7.0 Administration Guide Volume 2*.

8. Continue with “Mapping WebLogic Resources to NetPoint Resources” on page 131.

## Mapping WebLogic Resources to NetPoint Resources

The NetPointResourceMap.conf file contains mappings of WebLogic resources to NetPoint resources. These mappings allow NetPoint and WebLogic to recognize each other’s resource definitions. Netpoint recognizes only URLs, whereas each WebLogic resource has different set of elements associated with it. By mapping these resource elements to URLs and operations, all types of resources can be protected through NetPoint.

---

**Note:** The NetPointResourceMap.conf file is used only for the BEA Application Server integration, *not* for BEA Portal Server integration

---

After defining your NetPoint resources, policy domain, and so on, you need to be sure that the WebLogic resources that you want to protect will correspond to the resources that you defined in NetPoint.

### To map WebLogic Resources to NetPoint Resources

1. Locate the NetPointResourceMap.conf file. in the directory where NetPoint Security Provider for WebLogic is installed.
2. Review the format of this file in “NetPointResourceMap.conf File Format” on page 132.
3. Edit the file using information on NetPoint resource type definitions in “Setting Up WebLogic Policies in NetPoint” on page 117.
4. Continue with “Preparing the WebLogic Environment” on page 135.

## NetPointResourceMap.conf File Format

Resources that have mapping entries in this file are the only ones protected. Resources that do *not* have a mapping entry in this file are allowed access by default.

The format of entries in the NetPointResourceMap.conf file is as follows:

*Weblogic Resource Type:Netpoint Resource Type: enabled|disabled:URL prefix:URL pattern:Operation*

- **Weblogic Resource Type**—The WebLogic resource type. For example: <url>.
- **NetPoint Resource Type**—The NetPoint resource type that is mapped to the WebLogic resource type. For example: wl\_url.
- **enabled|disabled**—If enabled, all resources of the specified WebLogic resource type are protected by NetPoint. If disabled:
  - All users can access the resource if the status is disabled,allow.
  - No users can access the resource if the status is disabled,deny.
- **URL prefix**—The WebLogic elements that form a NetPoint URL prefix under which all resources are protected. This URL prefix is specified in a NetPoint policy. Each element in the URL prefix is a type of resource. For example, the following URL prefix can be used for EJBs:

*application/module/ejb*

where *application* and *module* are a specific WebLogic application and module.

- **URL pattern**—The WebLogic elements that form a more granular NetPoint URL pattern than is specified by a URL prefix. This a pattern is specified in a NetPoint policy. For example, to control access to users based on a particular method, you would specify:

*methodInterface/method*

- **Operation**—This maps to a WebLogic resource element such as HTTPMETHOD. If you specify a value in angle brackets (“<”), the policy returns the matching string. If you omit the brackets, the policy returns the value associated with the parameter.

**Figure 5** NetPointResourceMap.conf

```
#####  
# This file contains mapping of weblogic resources to Netpoint resources.  
# Netpoint only understands url representation whereas each weblogic resource has  
# different set of elements associated with it. So, by mapping these resource  
elements
```

```

# to url & operation all kinds of resources can be protected through Netpoint.
#
# The format of entries is as below.
# Weblogic Resource Type:Netpoint Resource Type: enabled/disabled:URL prefix:URL
pattern:Operation
# e.g. <url> : w1_url : enabled : contextPath : uri : httpMethod
# If the resource is configured disabled, then the default action can be
configured.
# e.g. disabled,allow
#
# If the operation doesn't come from weblogic resource field, and is fixed then
# it can be configured by putting the value between <> . For example <execute>
# If the value for operation is not specified then it defaults to "<default>"
#
# Leading & trailing white spaces in the fields are allowed. Blank lines are
allowed.
# Comments can be put by starting the line with #
#####
##### COMMONLY used resources #####

# HTTP resource. Available keys: application, contextPath, uri, httpMethod,
transportType
<url>:w1_url:enabled:contextPath:uri:httpMethod

#ejb - EJB resource. Available keys: application, module, ejb, method,
methodInterface, signature
# signature is ingorned here for performance reasons. You can include it if you
want to.
<ejb>:w1_ejb:enabled:application/module/ejb:methodInterface/method:<execute>

# web resource. Available keys: application, uri, webResource, httpMethod,
transportType
# This resource is deprecated by BEA in WLS 8.1. <url> replaces this resource type.
# You can enable it if you want to. Pls. refer to WLS 8.1 documentation for further
details.
<web>::disabled,deny:uri:webResource:httpMethod

# Server resource. Available keys: server, action
# Typically server=<wls server name>
<svr>:w1_svr:enabled:server::action

```

```

# Admin resource. Available keys: category, realm, action
# Typically for admin console category=Configuration. realm is ignored in default
configuration.
<adm>:wl_admin:enabled:category::action

##### LESS used resources #####

# JDBC resource. Available keys: resourceType, resource, action
#<jdbc>:wl_jdbc:enabled:resourceType:resource:action
<jdbc>::disabled,allow:::
# JMS resource. Available keys: destinationType, resource, action
#<jms>:wl_jms:enabled:destinationType:resource:action
<jms>::disabled,allow:::
# JNDI resource. Available keys: path, action
#<jndi>:wl_jndi:enabled:path::action
<jndi>::disabled,allow:::

```

---

**Note:** JNDI, JDBC, JMS resource protection is disabled by default. These can be enabled in `NetPointResourceMap.conf`.

---

The tool can be used to create policies automatically only for Web and ejb resources. It cannot be used for JNDI, JDBC, JMS, and other types of resources. These need to be created by hand in the NetPoint Access System Console. To find the URL pattern and operation used for the policy, set the log level to debug for the WebLogic SSPI package and look into the logs for the string “Entering OblixDatabase.isProtected for”. For example:

```

Entering OblixDatabase.isProtected for resource Type=wl_jndi,
isEnabled=true, URL=/weblogic/jms/
MessageDrivenBeanConnectionFactory, operation=lookup

```

For this example, a resource type `wl_jndi` needs to be first created in NetPoint Access System Console with a resource operation of “lookup”. Then policy needs to be created for the url `/weblogic/jms/MessageDrivenBeanConnectionFactory` (or parts of it such as `/weblogic/jms`) in the weblogic policy domain. For more information about protecting resources with NetPoint, see the *NetPoint 7.0 Administration Guide Volume 2*.

## Preparing the WebLogic Environment

The following procedure describes how to configure the WebLogic environment so that the NetPoint Security Provider is recognized by the WebLogic Server.

### To prepare the environment

1. Copy the mbean jar file from one of the following locations:

Copy from  
*install\_dir/oblix/lib/mbeantypes*  
to  
*WebLogic\_Home/server/lib/mbeantypes*

---

**Note:** If you are using WebLogic 8.1, copy *wl8NetPointSecurityProviders.jar*. If you are using WebLogic 7.0 SP2 and later, copy *wl7NetPointSecurityProviders.jar*.

---

2. Copy the files below from your *NetPoint\_install\_dir* to your WebLogic domain folder:

*NetPointProvidersConfig.properties*  
*NetPointResourceMap.conf*—only for the Application Server domain

3. Ensure that the following Admin credentials are set in *clear text* in the *NetPointProvidersConfig.properties* file:

*OB\_AdminUserName=admin*  
*OB\_AdminUserCreds=password*

If the *NetPointProvidersConfig.properties* file has a clear text password, the SSPI reads in the password, encrypts it, and re-writes the properties file with the encrypted password.

---

**Note:** *NetPointProvidersConfig.properties* file formatting is lost when NetPoint re-writes the file with the encrypted password. You may want to save a copy of the *NetPointProvidersConfig.properties* file. Also, ensure that all parameters are correctly filled as mentioned in “*NetPointProvidersConfig.properties*” on page 152.

---

You complete the next step if the NetPoint SSPI talks to a WebPass that is protected by a WebGate. Otherwise, skip to step 5.

4. **WebPass Protected by WebGate**—Complete the activities listed below when the NetPoint SSPI talks to a WebPass protected by a WebGate:
  - a) In the *NetPointProvidersConfig.properties* file, ensure that *OB\_WebPassIsProtected* is set to true. The *OB\_CookiePath* and *OB\_CookieDomain* parameters are configured correctly.

- b) Open the WebGateStatic.lst file in:  
`WebGate_install_dir\access\obl原因\apps\webgate`  
where `WebGate_install_dir` is the directory where WebGate is installed.
- c) Set `IPValidation = false`

---

**Note:** If you want to set `IPValidation = true`, check the `IPValidationExceptions` list for the IP address.

---

- d) Restart the Web server.

---

**Note:** Ensure that the security level in this authentication scheme is the same level or a lower level than the one specified in the WebLogic authentication scheme.

---

Next, you need to determine if the machine hosting WebPass is running SSL. If it is, complete step 5. Otherwise, skip to step 6.

- 5. **WebPass Host SSL-Enabled**—Determine if the machine hosting WebPass is running SSL, and if so, complete the following steps:

- a) Open the `NetPointProvidersConfig.properties` file and set `OB_WebPassSSLEnabled = True`.
- b) Obtain the CA certificate from the certificate authority to which the Web server hosting the WebPass or WebGate running in SSL mode has registered, and place it in `ca.cer` file.
- c) Use the `keytool` in `JAVA_HOME\bin` or `JAVA_HOME\jre\bin` to add the following ca certificate to cacerts keystore present in:

`JAVA_HOME\jre\lib\security` folder for weblogic jdk

```
keytool -import -alias ca -file ca.cer -keystore
JAVA_HOME\jre\lib\security\cacerts
```

- 6. Add the following environment variables in the WebLogic Server startup script *before* the command that starts the server:

Add the following to the CLASSPATH:

```
install_dir/obl原因/lib/wlNetPoint.jar
install_dir/obl原因/lib/bcprov-jdk14-125.jar
install_dir/obl原因/lib/xerces.jar
install_dir/obl原因/lib/jobaccess.jar
```

7. Add the following environment variables in the WebLogic Server startup script *before* the command that starts the server:

**Windows**—Add the following to PATH:

*install\_dir\oblix\lib*

**Solaris and Linux**—Add the following to LD\_LIBRARY\_PATH:

*install\_dir/oblix/lib*

**HP-UX**—Add the following to SHLIB\_PATH:

*install\_dir/oblix/lib*

**Portal Domain**—The CLASSPATH and PATH variables should be added just after the SAVE\_JAVA\_OPTIONS environment variable in the startWebLogic.cmd script (On Unix, it is the startWebLogic.sh script).

8. Remove the boot.properties file from the WebLogic domain directory.

This will cause the startWebLogic script described in the next step to prompt for username and password.

9. In the WebLogic domain directory, start the WebLogic Server using the appropriate startup script:

**Windows**—This command is startWeblogic.cmd

**Unix**—This command is startWeblogic.sh

Using the WebLogic 8.1 Domain Configuration Wizard, you can create instances of a new WebLogic 8.1 domain, for example, *mydomain*, and a new WebLogic 8.1 server, for example, *myserver*. You can also create instances of a new WebLogic 8.1.3 Portal domain, for example, *portalDomain*, and a new WebLogic 8.1.3 portal server, for example, *portalServer*.

10. Setup a NetPoint Realm that uses NetPoint security providers, as follows:

- a) Open a new console window and set the Weblogic environment by executing setEnv.cmd.

**Unix**—Source the setEnv.sh script present in the server domain directory.

**Portal Domain**—Use the setDomainEnv.cmd script (on Unix it is the setDomainEnv.sh script).

- b) Run the appropriate script, below, and ensure that it has the correct username, password, and URL values:

**Windows**—*install\_dir/setupNetPointRealm.cmd*

**Unix**—*install\_dir/setupNetPointRealm.sh*

**Portal Domain**—Run the script with parameter “portal”.

**WLS 7.0**—The script does not work and Realm must be set manually.

- c) Log in to the WebLogic Admin Console, navigate to *Domain* > Security > Realms and:
  - Verify that NetPoint realm is set as default.
  - Verify that the security providers are set properly in the NetPoint Realm.
- d) **Script Fails**—If the script fails, you must manually add the NetPoint security realm as described below:
  - Go to *Domain* > Security > Realms and select “Configure a new Realm”.
  - For the option “Check Roles and Policies for”, ensure that “All Web Applications and EJBs” is selected.
  - Navigate to Providers > Authentication, and configure a new NetPoint Authenticator and Identity Asserter.
 

**Identity Asserter**—Select the Token Type ObSSOCookie and in the Details tab, uncheck “Base64Decoding Required”.

**Portal Domain**—Set the control flag of NetPoint Authenticator to OPTIONAL and also configure a Default Authenticator.
  - Navigate to Providers > Authorization and configure a new NetPoint Authorizer (for the portal domain, only configure a Default Authorizer).
  - Navigate to Providers > Role Mapping and configure a new NetPoint Role mapper (for the portal domain, only configure a Default Role mapper).
  - Navigate to Providers > Credential Mapping and configure a new Default Credential mapper.
  - Navigate to *Domain* > Security and select this realm as the default realm.

You complete step 11 only when configuring a Portal Server domain. Otherwise, skip to step 12.

- 11. Portal Server Domain**—Complete the steps below to configure a Portal Server domain:
  - a) Restart the server using the same WebLogic credentials that were used earlier.
  - b) In the WebLogic Server Console, navigate to *Domain* > Security > Realms > NetPointRealm > Providers > Authentication, and:
    - Remove the Default Authenticator.
    - Change the control flag for NetPoint Authenticator to REQUIRED.

- c) Using the NetPoint Group Manager, create a group in NetPoint that maps to the Admin role in the BEA WebLogic Server *and* contains all the administrators for the BEA Portal.

For example:

*BEA\_Administrators*

- d) Create a user (portaladmin) and add it to the *BEA\_Administrators* group; later you login as this user (portaladmin) when restarting the server.
- e) In the WebLogic Server Console Admin Console, navigate to Security > Realms > NetPointRealm and:
  - Click Groups to display all NetPoint groups.
  - Search for the BEA Admin group that was created above using a wild card search if you like.
  - Copy the group name.
- f) Click Global Roles > Admin role > Conditions tab and:
  - Add a Role Condition where the caller is a member of the group.
  - Paste in the group name you copied.
- g) Change the role condition from “and” to “or”, then click Apply.
- h) Repeat this procedure for the PortalSystemAdministrator role.

---

**Note:** Other BEA roles can be mapped to NetPoint groups/users. When you restart the WebLogic Server, it is important that you are logged in as a user in the NetPoint group associated with the BEA Admin role.

---

## 12. Restart the WebLogic Server.

The next time you log in to the WebLogic console, provide NetPoint administrator credentials. You will be authenticated using the NetPointRealm.

## 13. If you are using identity assertion as the authentication mechanism that protects Web applications:

- a) Install a NetPoint WebGate on the proxy Web server. See “Authentication for Web-Only Resources” on page 106 for an illustration of this type of installation.
- b) Configure the NetPoint policies that protect the Web applications to use HTTP as the resource type instead of wl\_url.

---

**Note:** There is one exception to the resource type configuration. The WebLogic administration console always uses form login. The /console policy must use the resource type wl\_url.

---

14. If anything other than a form-based NetPoint authentication scheme protects the policies configured with the HTTP resource type, configure a challenge redirect parameter to redirect the user to another Web server that has WebGate installed.

---

**Note:** If you do not complete this step, the user will have to refresh the browser to access the desired page because the ObSSOCookie set by the WebGate in the HTTP request has *not* yet been sent to the WebLogic server.

---

15. Continue with the following procedures, as needed:
  - “Configuring the COREid Server” on page 140
  - “Configuring Multiple WebPass Instances” on page 141

## Configuring the COREid Server

Next, you complete the procedure below to configure the COREid Server.

### To configure the COREid Server

1. Open the oblixappparams.xml file and set the searchstringMinimumLength to zero:

```
COREid_install_dir\identity\oblix\apps\common\bin\oblixappparams.xml
<NameValuePair ParamName="searchstringMinimumLength" value="0"/>
```

where *COREid\_install\_dir* is the directory where you installed NetPoint COREid Server.

2. Open the groupservcenterparams.xml file and set the groupMemberSearchStringMinimumLength to zero:

```
COREid_install_dir\identity\oblix\apps\groupservcenter\bin\groupservcenter
params.xml
<NameValuePair ParamName="groupMemberSearchStringMinimumLength"
value="0"/>
```

3. Restart the COREid Server.

The next step must be completed after COREid System setup.

4. From the COREid System Console, create an administrator with the required View and Delegated Administration rights.

---

**Note:** This administrator should be the one used for “OB\_AdminUserName” parameter in the NetPointProvidersConfig.properties. For more information about configuring administrators, see the *NetPoint 7.0 Administration Guide Volume 1*.

---

## Configuring Multiple WebPass Instances

NetPoint uses failover to maximize performance and provide uninterrupted service to end users. Failover redirects requests when a server fails. You may want to configure multiple WebPass instances for the failover purposes.

This section assumes that you have already installed more than one instance of WebPass for the NetPoint Connector. See the *NetPoint 7.0 Deployment Guide* for more information on failover.

### To configure multiple WebPass instances

1. Open the `NetPointProvidersConfig.properties` file in the WebLogic domain directory.
2. Enter the WebPass fully-qualified hostname with the domain name and port number using a comma-separated list.

For example:

```
# NetPoint WebPass webserver host name and port number
```

```
OB_WebPassHost=foo.domain.com,bar.doman.com
```

```
OB_WebPassPort=81,80
```

In the above example, the valid WebPass *host:port* combinations are:

```
o foo.domain.com:81  
o bar.domain.com:80
```

## Configuring SSO for the Portal Server

NetPoint 7.0.3 supports integration with the WebLogic Portal Server.

To enable SSO between Portal Web Applications and NetPoint protected resources, the Portal Web Application must be set up for ObSSOCookie handling. The prerequisite to support SSO for the Portal Web Application is that it should be using a form-based login portlet for authentication. The “sampleportal” Web Application module a part of “portalApp” Web Application, which is shipped with Weblogic Portal 8.1.3 and is considered as an example.

Following is an outline of the procedures you need to complete to set up SSO for the Portal Web Application module.

### Task overview: Configuring SSO for the Portal Server includes

1. “Configuring web.xml” on page 142
2. “Configuring the login jsp used by the Login Portlet” on page 142
3. “Copying ObLoginFilter.class in the WEB\_INF/classes” on page 144

4. “Completing Setup” on page 144
5. “Testing SSO for the Portal Server” on page 145

## Configuring web.xml

You need to include filter-related nodes at the start of other filter nodes.

---

**Note:** The mapping mentioned in the filter and the POST action url set in the form should be the same. The only difference is that the action url will include the context root too and the Oblogin\_validate.jsp name will be present.

---

### To add filter-related nodes

1. Locate the Portal Application’s Web module’s WEB-INF/web.xml.

2. Add the following filter related nodes *at the start of* other filter nodes:

```
<!-- Login Servlet Filter, required for SSO between Portal and
Netpoint -->
    <filter>
        <filter-name>OblixLoginFilter</filter-name>
        <filter-class>ObLoginFilter</filter-class>
    </filter>
```

3. Add the following filter mapping node *at the start of* other filter mapping nodes:

```
<filter-mapping>
    <filter-name>OblixLoginFilter</filter-name>
    <!-- Need to configure below mapping correctly to invoke the
Oblix Login filter -->
    <url-pattern>/portlets/login_validate/*</url-pattern>
</filter-mapping>
```

## Configuring the login jsp used by the Login Portlet

You need to configure the following in the login jsp used by the login portlet.

### To configure the login jsp for the Login Portlets

1. Include following at the start of the login jsp page.

```
<%@ page import="com.bea.portlet.PostbackURL,
com.bea.netuix.servlets.controls.content.JspContentContext" %>
    <% // Set this url as per your setting %>
    <%@include file="/portlets/NetPointPortalSSO.jsp" %>
<%
```

```
// Included to get the Base URL for redirection after
Authnetication.
JspContentContext jspContentContext =
JspContentContext.getJspContentContext(request);
PostbackURL url = jspContentContext.getBaseUrl(request,
response, "");
%>
```

2. Set the form's action url as following:

```
<form method="post" action="/sampleportal/portlets/
login_validate/Oblogin_validate.jsp" type="POST">
```

---

**Note:** The action URL needs to start with the context root included.

---

3. Set the user input fields used in login.jsp to get the username and password to username and password, respectively.

4. Include a new variable in the login input form:

```
<tr>
  <td align="left"> <input type="hidden" name="targeturl"
value=<%= url %> > </td>
</tr>
```

5. If your form allows logout, set the logout url to Oblogout.jsp. Else in your logout logic include following code to kill the ObSSOCookie

```
<%@ page
import="com.oblix.weblogic.configuration.NPConfiguration"%>
<%@ page import="com.oblix.weblogic.logging.ObDebug"%>
<%
    // begin block to kill obSSOCookie
    // Check if the user has obSSOCookie
    ObDebug.getInstance().debug("Inside logout.jsp");
    Cookie[] cookies = request.getCookies();
    if ( cookies != null ){
        String obSSOCookie = null;
        for (int i = 0; i < cookies.length; i++) {
            if (cookies[i].getName().equals("ObSSOCookie")) {
                obSSOCookie = cookies[i].getValue();
                // if obSSOCookie is not null and is not 'loggedout' then
                // kill it by making it loggedout
            }
        }
        if (obSSOCookie != null && ! obSSOCookie.equals("") && !
obSSOCookie.equals("loggedout")){

            Cookie killedSSOCookie = new Cookie("obSSOCookie", "loggedout");
```

```

        String cookieDomain = NPConfiguration.getCookieDomain();
        if(cookieDomain != null && cookieDomain.length() > 0)
            killedSSOCookie.setDomain(cookieDomain);
        killedSSOCookie.setPath("/");
        response.addCookie(killedSSOCookie);
    ObDebug.getInstance().debug("Logout jsp - ObsSOCookie set to
    loggedout with domain [" + cookieDomain + "]");
    }
    break;
    }
    }
}
// end block to kill ObsSOCookie
%>

```

## Copying ObLoginFilter.class in the WEB\_INF/classes

The ObLoginFilter.class expects that the name of user input fields used in login form are “username” and “password”. However, other names can be used.

### To use other names

1. Modify the file provided.
2. Compile the file and include it under the WEB-INF/classes folder.
3. Proceed to “Completing Setup” on page 144.

## Completing Setup

Use the following procedure to complete the setup process for this implementation.

In the sampleportal example included under the “PortalApp” application that is shipped as an example with Weblogic 8.1 SP3 we had created the “login\_validate” folder under the “portlets” folder present under the context root and copied the Oblogout.jsp, NetpointPortalSSO.jsp just below “portlets” folder.

---

**Note:** The files mentioned above are included in installation directory under: oblix/examples/src/webapp/portalApp/sampleportal.

---

Except for Oblogout.jsp, Oblogin\_validate.jsp and NetPointPortalSSO.jsp other jsp files need to be configured for each application.

### To complete setup

1. Copy Oblogout.jsp under context root of the application.
2. Copy NetpointPortalSSO.jsp under context root of the application.
3. Under context root of your application module, create a folder “login\_validate” and copy the Oblogin\_validate.jsp.

This file’s contents is displayed only when the filter is *not* invoked.

4. Continue with “Testing SSO for the Portal Server” on page 145.

## Testing SSO for the Portal Server

You can test the examples provided in the following directory:

*NetPoint\_Security\_Provider\_install\_dir/examples*

These samples allow you to test single sign-on for Web applications and EJBs either using or not using identity assertion for authentication.

There are readmes in the example directories. These readmes provide instructions for testing different types of resources.

**For the Portal Server**—The examples\src\webapp\portalApp\sampleportal does *not* contain a full sample that can be deployed.

### To test SSO for the Portal Server

1. Configure the *WebLogic\_install\_dir\samples\domains\portal* domain to use the NetPoint Security Provider.

2. Locate the deployed sampleportal example:

*WebLogic\_install\_dir\samples\portal\portalApp\sampleportal*

3. Copy/replace the files below:

**From—**

*NetPoint\_Security\_Provider\_install\_dir/examples\src\webapp\portalApp\sampleportal*

**To—**

*WebLogic\_install\_dir\samples\portal\portalApp\sampleportal* as follows:

login.jsp	> \portlets\login.jsp
NetPointPortalSSO.jsp	> \portlets\ NetPointPortalSSO.jsp
Oblogout.jsp	> \portlets\ Oblogout.jsp
web.xml	> \WEB-INF\web.xml
ObLoginFilter.class	> \WEB-INF\classes\ ObLoginFilter.class
Oblogin_validate.jsp	> \portlets\login_validate\Oblogin_validate.jsp

4. Either restart the Portal Server or redeploy the PortalApp example.

# Context-Specific Authorization

You can configure the NetPoint Security Provider for WebLogic to perform context-specific authorization, in which the authorization scheme uses context data present in the HTTP servlet request for Web applications to determine if a user is allowed access. For example, the authorization scheme can determine that a request must come from a particular IP address, such as the user's home or work machine. The access decision can be based on other factors such as rules. For example, you can allow access only if the user belongs to the Engineering role.

See the *NetPoint 7.0 Administration Guide Volume 2* and the *NetPoint 7.0 Developer Guide* for details on context-sensitive authorization schemes.

Information from a user's role and the ContextHandler is sent to the Access Server based on the configuration of a custom authorization scheme. The ContextHandler is an object that WebLogic passes to the SSPI. The Access Server passes this information to a custom authorization plug-in that makes the authorization decision. It supports three formats for the user parameter field in the authorization scheme definition:

- **RA\_roles**—Roles that the user belongs to appear as a comma-separated string.
- **RA\_http.method**—This executes the *method* on an `HttpServletRequest` object and returns a string value. For example, `RA_http.getRemoteAddr()` gets the IP address of the machine where the user sent the request.
- **RA\_http.session.method**—This executes the *method* on an `HTTPSession` object and returns string values. For example, `RA_http.session.getAttribute(myattribute)` gets the value of the session attribute named *myattribute* and returns it as a string. For example, if you want to provide a coupon to a user who has \$1,000 of purchases in a shopping cart, you can detect the spending level on a session attribute.

Context-sensitive authorization can be done for EJBs based on the method parameter values apart from the roles. This is done using the reverse action functionality provided by the Access Server. A custom authorization scheme is required for this purpose. You can demonstrate this for a stateless EJB using the procedure below, which limits access to the buy method for the BEAS stock only.

## To implement an example

1. Copy the shared library `req_context`, as follows:

**From**—`install_dir/examples/src/webapp/contextAuthz`

**To**—Access Server host, `AccessServer_install_dir/oblix/lib`

2. Navigate to the Access System Console > Access System Configuration > Authorization Management, then click the Add button.

3. In the Shared Library field, enter the path to the req\_context.  
For example:  
c:\NetPoint\access\oblix\lib\req\_context
4. In the User Parameter field, add RA\_ejb.Parameter1.
5. In the Required Parameter field, add the following name value pair.  
paramName\_1     ejb.Parameter1  
paramValue\_1     BEAS
6. After saving this scheme, restart the Access Server.
7. Create an authorization definition using this authorization scheme in the domain used for SSPI.
8. Modify the policy for the EJB buy method by setting the authorization rule to the above authorization definition.
9. Run the EJB client.

You should get the following output, where the buy of BEAS shares goes successful but the buy of MSFT shares fail due to access control.

run:

```
[java]
[java] Beginning statelessSession.Client...
[java]
[java] user: admin
[java] Creating a trader
[java] Buying 100 shares of BEAS.
[java] Buying 200 shares of MSFT.
[java] There was an exception while creating and using the
Trader.
[java] This indicates that there was a problem communicating
with the server: java.rmi.AccessException:
[EJB:010160]Security violation: User: 'admin' has insufficient
permission to access EJB: type=<ejb>,
application=_appsdir_ejb20_basic_statelessSession_ear,
module=ejb20_basic_statelessSession.jar, ejb=statelessSession,
method=buy, methodInterface=Remote,
signature={java.lang.String,int}.
[java]
[java] End statelessSession.Client...
[java]
```

A custom plug-in similar to this can be written in C to implement the business logic according to your needs.

There is an example of context-specific authorization provided with the security Web application sample in the installation directory. Details for how to configure the authorization scheme is provided in the readme.

## Audit Files

To enable auditing, you need to add an auditing provider in the WebLogic security realm. There is a default provider installed when you install WebLogic.

The default auditing provider writes the audit records to the following file:

```
WebLogic_domain_directory/server/DefaultAuditRecorder.log
```

where *WebLogic\_domain\_directory* is the name of the WebLogic domain and *server* is the server name.

WebLogic audits all authentication and authorization successes and failures. The NetPoint Security Provider for WebLogic audits the reason for any failures. The failure logs appear before the WebLogic failure log.

## Debug Log Files

WebLogic writes debug logs to the following files:

```
WebLogic_domain_directory/server/server.log
```

where *WebLogic\_domain\_directory* is the name of the WebLogic domain and *server* is the server name.

You can configure the log level from the WebLogic administration console. For debugging, set the configuration parameter `ObDebugMode=true` in the `NetPointProvidersConfig.properties` file. See “`NetPointProvidersConfig.properties`” on page 152 for details. The change is takes effect after 60 seconds. Debug logs are written to the *server.log* file. You can also configure debug logs to print to stdout from the WebLogic administration console.

For more information about a log from the catalog, use WebLogic’s `CatInfo` utility. Set the WebLogic environment by running the `setdomainEnv.cmd` (or `.sh` for Unix), and then run:

```
java weblogic.il8ntools.CatInfo -id message ID
```

This command also lists the cause action details for the log message.

# User Creation/Deletion and Group Creation

The NetPoint SSPI uses workflows defined in the NetPoint COREid System to create users and groups and delete users. The data available to be passed in a workflow request is limited by the SSPI interface of WebLogic, as follows:

- **Create User**—The Userid, Password, and Description parameters are available to the SSPI while making a Create User request.
- **Delete User**—The only attribute that is available while deleting a user is Userid.
- **Create Group**—The parameters available while creating a group are Name and Description.

It is possible to define a workflow that uses values for these attributes. It is also possible to send constant values for more attributes as shown, in the following sample workflow definition in Table 6, “Workflow Fields and Values,” on page 150.

**Table 6** Workflow Fields and Values

Workflow Field	Sample Workflow Values
Workflow Name	Name generated by the NetPoint BEA SSPI Create User Workflow.
Workflow Type	Create User
Workflow DN	obworkflowid=wfqs20020806T0907402920,obcontainerId=workflowDefinitions,OU=Oblix,OU=Company,DC=qalab-vduong,DC=oblix,DC=com
Workflow Status	Enabled
Description	Workflow generated for NetPointBEASSPI
Target	Company:OU=Company,DC=qalab-vduong,DC=oblix,DC=com
Workflow Domain	OU=Company,DC=qalab-vduong,DC=oblix,DC=com
Workflow Steps	Step 1: Name: Initiate Attribute Name: LoginID (Required) Attribute Name: Password (Required) Attribute Name: Name (Required) Participant: admin Step 2: Name: Enable Entry Condition: 1. true:false

Following are the corresponding parameters from the NetPointProvidersConfig.properties file:

```
OBWebPass.CreatUserWorkFlowID=wfqs20020806T0907402920,
obcontainerId=workflowDefinitions, OU=Oblix,
OU=Company,DC=qalab-vduong,DC=oblix,DC=com
```

```
OBWebPass.CreatUserWorkFlowDomain=OU=Company,DC=qalab-vduong,DC=oblix,DC=com
```

In this file, \$UID\$ and \$PASSWORD\$ denote value of login attribute and password, respectively. The placeholders are passed to the workflow *as is* and are written to the user profile. At runtime, both \$UID\$ and \$PASSWORD\$ are replaced with values obtained for the login attribute and password.

```
OBWebPass.CreatUserWorkFlowNumOfFields=3
```

```
OBWebPass.CreatUserWorkFlowAttrName_1=cn
```

```
OBWebPass.CreatUserWorkFlowAttrValue_1=Name of $UID$
```

```
OBWebPass.CreatUserWorkFlowAttrName_2=uid
```

```
OBWebPass.CreatUserWorkFlowAttrValue_2=$UID$
OBWebPass.CreatUserWorkFlowAttrName_3=userPassword
OBWebPass.CreatUserWorkFlowAttrValue_3=$PASSWD$
OBWebPass.CreatUserWorkFlowComment=Added user $UID$ from
webLogic portal server.
```

If the workflow is modified to use different attributes, the preceding sample lines in the `NetPointProvidersConfig.properties` file need to change. If workflow is modified to use another attribute, the DS attribute name must be specified in this file.

For example, if you change the first attribute from `cn Name` to `cn Mail` (that is `obmail` in DS) then do the following:

```
OBWebPass.CreatUserWorkFlowAttrName_1=obmail
OBWebPass.CreatUserWorkFlowAttrValue_1= $UID$@company.com
```

Similarly, if this is a new parameter being added to the workflow, you need to increase the number of fields and add two new lines for attribute:

Increase the number of fields as shown below:

```
OBWebPass.CreatUserWorkFlowNumOfFields=4
```

Add two new lines, as indicated below:

```
OBWebPass.CreatUserWorkFlowAttrName_4=obmail
OBWebPass.CreatUserWorkFlowAttrValue_4=$UID$@company.com
```

The same approach is to be followed during group creation and user deletion. For group deletion, determination of the workflow to be used is made by NetPoint. The delete group operation requires only the Group DN, which is obtained from the Portal Admin interface at runtime.

## Configuration Files

The following configuration files are required for the integration of NetPoint and WebLogic:

- “`NetPointProvidersConfig.properties`” on page 152
- “`NetPointWeblogicTools.properties`” on page 160

## NetPointProvidersConfig.properties

Table 7 describes all common configuration items. There are also items written by the installer when you select the Typical/Advanced install of the NetPoint Security Provider for WebLogic. A sample file containing all of the parameters in the

A sample file containing all of the parameters in the SampleNetPointProvidersConfig.properties is located in:

*NetPoint\_Security\_Provider\_install\_dir/examples*

where *NetPoint\_Security\_Provider\_install\_dir* is the directory where the NetPoint Security Provider for WebLogic is installed. Table 7 describes these parameters.

**Table 7** NetPointProvidersConfig.properties

Parameter	Description and Value
ObDebugMode	Specifies whether NetPoint debugging information is recorded in the WebLogic log files. Default=false
ObRoles.Cache.TTL	Specifies the length of time for which security roles are cached. Default=60 seconds
ObRoles.Cache.CleanupSchedule	Specifies when expired items in the Roles cache are flushed. This is to reclaim memory. Default=60 seconds
ObAuthorization.ActionOnUnprotectedResource	Specifies default access to resources not protected by NetPoint. Default=allow
Ob_InstallDir	Specifies the installation directory for the NetPoint Security Provider.
OB_AuthnSchemeResourceTypeName	The NetPoint Security Provider uses a special policy to authenticate users in WebLogic. The resource type specified on this parameter is used in this special authentication policy. This resource type contains a resource with a URL prefix of /Authn/Basic. Value: Name of the resource type.

**Table 7** NetPointProvidersConfig.properties

OB_AuthnSchemeResourceName	The NetPoint Security Provider uses a special policy to authenticate users in WebLogic. The resource specified on this parameter is used in this special authentication policy. This resource contains a URL prefix of /Authn/Basic. Value: Name of the resource.
ObAuthentication.Anonymous.ResourceName	The name of the resource used in the policy for anonymous authentication.
OB_AuthnSchemeOperation	The operations specified on the resource for this policy. Example: LOGIN
ObAuthentication.LoginIdParam	The userID challenge parameter that is used in the authentication scheme for the policy. This value is the same as the userID challenge parameter specified in the credential_mapping plug-in. Example:userid
ObAuthentication.passwordParam	The password challenge parameter that is used in the authentication scheme for the policy. This value is the same as the password challenge parameter specified in the validate_password plug-in. Example:password
OB_AuthzActionType	The authorization action that retrieves the user's login ID from the ObSSOCookie. Example: WL_REALM
OB_AuthzActionName	The action that retrieves the user's login ID from the ObSSOCookie. Example: uid
ObFormAuthenticationSSO.DummyUserName	This dummy user name is used with a form login when single sign-on is configured. In this scenario, there is no WebGate on the proxy HTTP server. See "Completing Advanced Installation" on page 115 for details.
ObWebAppResourceTypes	WebLogic resource types that are used for Web applications.

**Table 7** NetPointProvidersConfig.properties

ObRoles.ResourceType	This parameter specifies the resource type defined for the policy that retrieves the user's role information. Example: wl_authen
ObRoles.ResourceName	This parameter specifies the URL prefix of the resource (not the resource type) defined for the policy that retrieves the user's role information. Example: /Authen/Roles
ObRoles.ResourceOperation	The operation specified on the resource in this policy. Example: LOGIN
ObRoles.ActionType	The action specified on the authorization rule for the policy that obtains the user's role. Example: WL_REALM
ObAuthorization.AbstainMapsTo	If you do not want to use the Abstain result of an isAuthorized call, you can map the result to allow or deny. See "Completing Advanced Installation" on page 115 for details.
ObAuthorization.OnDenyRedirectToUrl	If access to a resource is denied, then you can redirect the user to this page. This is Optional.
OB_LogLevel	The logging level that is recorded in the log file. Values are none, info, and debug. This is Optional.
OB_LogFileName	The file name for log messages. Default = install_dir/log. This is Optional.
OB_LogMilliseconds	The data/time format of log messages in the file specified with OB_LogFileName. When true, log messages are time formatted in milliseconds. Default =true. This is Optional.

**Table 7** NetPointProvidersConfig.properties

OB_WebPassHost	<p>The WebPass server host machine name. The host name must be fully qualified; for example, OB_WebPassHost=hostname.acme.com. To configure multiple WebPass instances for failover purposes, separate the names with a comma. For example: OB_WebPassHost=foo.domain.com, bar.domain.com</p> <p><b>Note</b> that the host name corresponds to the port number in the specified order. See the example in the Ob_WebPassPort description section below. This is Optional.</p>
OB_WebPassPort	<p>The port number of the host machine. To configure multiple WebPass instances, separate the port numbers with a comma. For example: OB_WebPassPort=80, 81</p> <p><b>Note</b> that the host name corresponds to the port number in the specified order. In the above example, the hostname:port number pairing is as follows: foo.domain.com:80 bar.domain.com:81</p> <p>For failover to work, all other variables such as user name, credentials and webgate protection must be the same. This is Mandatory.</p>
OB_WebPassIsProtected	<p>Values are true and false. If WebPass is protected, set value=true. This is Mandatory.</p>
OB_AdminUserName	<p>NetPoint requires the Admin username and password to make IdentityXML calls to the WebPass. This is Mandatory.</p>
OB_AdminUserCreds	<p>NetPoint requires the Admin username and password to make IdentityXML calls to the WebPass. Without the password the connector will not work.</p> <p><b>Note:</b> You need to enter a clear-text password, which the program will encrypt and re-write to the properties file after the first run. This is Mandatory.</p>

**Table 7** NetPointProvidersConfig.properties

OB_CookieDomain	The cookie domain specified in the NetPoint WebGate installer configuration. Needed if WebPass is protected. For example, .xyz.com. This is Mandatory.
OB_CookiePath	The cookie path specified in the NetPoint WebGate configuration. Needed if WebPass is protected. Default = /. This is Mandatory.
OB_WebPassSSLEnabled	Specifies whether WebPass needs HTTPS connection. Values are true and false. Default = false. This is Mandatory.
OB_UserAttr	The unique user identification (for example, uid). This is Mandatory.
OB_UserSearchAttr	The DN prefix for users from LDAP (for example, cn). This is Mandatory.
OB_GroupSearchAttr	The DN prefix for groups from LDAP (for example, cn). This is Mandatory.
OB_WebPassADDDomain	Optional. The domain of the Admin user. To be used in case of Active Directory Forest with multiple domains. For example, OB_WebPassADDDomain=ou=company,dc=qalab,dc=acme,dc=com The ADDomain must be the same as the default defined in the COREid Server. This is Optional.
OB_WebPassXPIRecordsReturned	Optional. The number of records to return for getUsers or getGroups. Default = return all. This is Optional.
OB_UserGroupsCache_enabled	Enables caching of list groups of which the user is a member. Values are true and false. Maintains a cache of all the groups a logged in user belongs to. This is Optional.
OB_UserGroupsCache_timeout	The timeout for cache of the list of groups for a user. The timeout is per user. This value should not be very high--if the user's group membership changes the new membership will only take affect at cache timeout. For example, a value of 3600 equates to 1 hour. This is Optional.

**Table 7** NetPointProvidersConfig.properties

OB_GroupMembersCache_enabled	Enables caching of list of groups and list of members in each group. Values are true and false. Stores members for each groups (not a frequently used cache). This is Optional.
OB_GroupMembersCache_timeout	Specifies the timeout for cache of list of groups and the list of members in each group. This is Optional.
OB_UserAttributesCache_enabled	Enables Caching of User Attributes. Values are true and false. This is Optional.
OB_UserAttributesCache_timeout	The timeout for the cache of user attributes. Timeout is for the whole cache. This is Optional.
OB_UserAttributesCacheElement_timeout	The timeout for the cached user attributes. The Timeout is per user. This is Optional.
OB_GroupAttributesCache_enabled	Enables Caching of Group Attributes. Values are true and false. This is Optional.
OB_GroupAttributesCache_timeout	The timeout for the cache of group attributes. Timeout is for the whole cache. This is Optional.
OB_GroupAttributesCacheElement_timeout	The timeout for the cached group attributes. The Timeout is per group. This is Optional.
OB_Keystore	Specifies the keystore file used by the SSPI connector when it makes SSL connections to HTTPS WebPass. The keystore contains the requestor's public and private key pairs, X.509 certificate, and certificates for Certificate Authorities trusted to certify responder servers. The keystore is managed using the JDK keytool. For example: <i>install_dir/oblix/config/jssecacerts</i> . This is Optional.
OB_KeystorePassword	The password for the keystore. This is Optional.
OB_UserTabId	For future use. Do not change the default. Default = Employees

**Table 7** NetPointProvidersConfig.properties

OB_GroupTabld	For future use. Do not change the default. Default = Groups
OB_NestedGroupsEnabled	<p>Values are true and false. The default is true. To improve GroupSrvCenter performance when nested groups are not used, set the value to false.</p> <ul style="list-style-type: none"> <li>• Nested groups will not be included in the search; the uniquemember attribute will not be requested in a group search when OB_NestedGroupsEnabled=false.</li> <li>• A value of true retrieves the uniquemember attribute in the group search, uses this for nested group computation, then removes it before the group is recorded.</li> </ul> <p>This is Optional.</p>
OB_DynamicGroupsEnabled	<p>Values are true and false.</p> <p>To improve GroupSrvCenter performance when you are not using dynamic groups, set the value to false. Dynamic groups will not be included in the search. This is Optional.</p>
OB_UserPasswordAttr	<p>User Password Attribute. Example: userpassword. This is Mandatory.</p>
OB_UserDescriptionAttr	<p>User Description Attribute. Example: description. This is Mandatory.</p>
OBWebPass.CreatUserWorkFlowID	WorkflowID for create user workflow. This is mandatory <i>if</i> user creation is supported.
OBWebPass.CreatUserWorkFlowDomain	WorkflowDomain for create user workflow. This is mandatory <i>if</i> user creation is supported.
OBWebPass.CreatUserWorkFlowNumOfFields	Number of attributes present in the create user workflow. This is mandatory <i>if</i> user creation is supported.
OBWebPass.CreatUserWorkFlowAttrName_{number} OBWebPass.CreatUserWorkFlowAttrValue_{number} pair	Name/Value pair for an attribute during user creation.
OBWebPass.CreatUserWorkFlowComment	Comment while creating a user.

**Table 7** NetPointProvidersConfig.properties

OBWebPass.DeactivateUserWorkFlowID	WorkflowID for delete user workflow. This is mandatory if user deletion is supported.
OBWebPass.DelUserWorkFlowNumOfFields	Number of attributes present in the delete user workflow. This is mandatory if user deletion is supported.
OBWebPass.DelUserWorkFlowAttrName_{number} OBWebPass.DelUserWorkFlowAttrValue_{number} pair	Name/ Value pair for an attribute during user deletion.
OBWebPass.DelUserWorkFlowComment	Comment while deleting a user.
OB_GroupIDAttr	Group Name Attribute. Example: cn. This is Mandatory.
OB_GroupDescriptionAttr	Group Description Attribute. Example: description. This is Mandatory.
OB_GroupUniqueMemberAttr	Group Uniquemember Attribute. Example: uniquemember. This is Mandatory.
OBWebPass.CreatGroupWorkFlowID	WorkflowID for create group workflow. This is mandatory if group creation is supported.
OBWebPass.CreatGroupWorkFlowDomain	WorkflowDomain for create group workflow. This is mandatory if group creation is supported.
OBWebPass.CreatGroupWorkFlowNumOfFields	Number of attributes present in the create group workflow. This is mandatory if group creation is supported.
OBWebPass.CreatGroupWorkFlowAttrName_{number} OBWebPass.CreatGroupWorkFlowAttrValue_{number} pair	Name/ Value pair for an attribute during group creation.
OBWebPass.CreatGroupWorkFlowComment	Comment while creating a group.

## NetPointWeblogicTools.properties

Table 8 describes the NetPointWeblogicTools.properties file located in:

*NPSPWL\_install\_dir/oblix/tools/npWLTools*

where *NPSPWL\_install\_dir* is the directory where the NetPoint Security Provider for WebLogic is installed.

This file contains information on the WebLogic policy domain.

**Table 8** SampleNetPointWebLogicTools.properties Parameters

Parameter	Description and Value
ObWLTTools.Debug	Creates a debug file. Default = true
ObWLTTools.LogFile	Creates a log file. Default = NetPointWeblogicTools.log
ObPolicyDomain.Name	The WebLogic domain name created by the NetPoint Policy Deployer. Default = SecuProvForWeblogic
ObPolicyDomain.Description	Description of the WebLogic Domain.
ObPolicyDomain.LoginAttribute	The return attribute in the action of an authentication policy that retrieves the user's login ID from the ObSSOCookie. The return attribute is necessary for NetPoint single sign-on. Default = uid
ObWLTTools.SetupInitialNetpointSSPI Policies	Sets up the initial NetPoint policies for WebLogic such as / Authen/Bank. Use the default value when you first run the NetPoint Policy Deployer tool. When you run the tool subsequently, change value to false. Default = true
ObWLTTools.DeployPolicy	If set to true, the tool reads the NetPointDeployPolicy.txt file and creates the policies in NetPoint.
ObWLTTools.UnDeployPolicy	Deletes policies. Default = false
ObWLTSDomain.Dir	The local directory where the WebLogic domain is located. Default = c:/bea/user_projects/mydomain

**Table 8** SampleNetPointWebLogicTools.properties Parameters

ObWLAuthenticationScheme. Name	The authentication scheme used for WebLogic. This should be created manually before running the tool. Default = NetPoint WebLogic Basic Over LDAP
ObWLLNoneAuthenticationScheme. Name	The AuthenticationNone scheme used for WebLogic. This should be created manually before running the tool. Default = NetPoint WebLogic Anonymous Authentication
ObWLWebResource. usingIdentityAssertion	Whether identity assertion is used to protect Web applications. Default=false.
ObWLWebResource.proxyPrefix	The string trimmed from the beginning of the URL that the user originally specifies, before the request is forwarded to the WebLogic Server. For example, if the URL <code>http://myWeb.server.com/weblogic/foo</code> is requested, the URL forwarded to the WebLogic Server is: <code>http://myWeb.server.com:7001/foo</code> The part of the request that is trimmed is what you specify on the <code>ObWLWebResource.proxyPrefix</code> parameter.

## Implementation Notes for Active Directory

The following are issues to consider when implementing the NetPoint Connector for WebLogic on Active Directory.

- “Configuring NetPoint Connector for WebLogic” on page 161
- “Setting a Domain in NetPointProvidersConfig.properties” on page 162

## Configuring NetPoint Connector for WebLogic

The steps to configure NetPoint Connector for WebLogic for an Active Directory Forest follow.

### To configure the NetPoint Connector for an Active Directory forest

1. In the NetPoint Access System Console, create a new Basic Over LDAP authentication scheme for a domain in the Active Directory Forest.

The base credentials that you specify in the Plugin(s) field *must* be the same as the search base that you specified in the directory server profile. See page 110.

You need to complete step 2 only if you did *not* create an administrator during pre-installation setup. Otherwise, skip to step 3.

2. Create a WebLogic administrator in NetPoint with View and Delegated Administration rights and ensure that the administrator's login identification is unique.
3. Specify the WebLogic administrator as the administrator for the Active Directory forest domain.

This domain must be the same as the one for which you created the authentication scheme in step 1. To do this, specify values for the `OB_WebPassADDomain` parameter in the `NetPointProvidersConfig.properties` file as described in Table 7 on page 152.

You can search for users in the parent domain but you *cannot* search for users in sibling or children domains.

---

**Note:** You do *not* need to create an administrator for every domain in an Active Directory Forest.

---

## Setting a Domain in `NetPointProvidersConfig.properties`

If you are running Active Directory using multiple domains, you must manually edit the `NetPointProvidersConfig.properties` file to include a value for the `OBWebPassADDomain` parameter. For example:

```
OBWebPassADDomain=dc=xyz, dc=acme, dc=com
```

The domain *must* be the same as the domain defined for the default directory server in the COREid System.

See the *NetPoint 7.0 Administration Guide Volume 1* for more information.

### To Prepare the BEA WebLogic Server 8.1.x

1. Ensure that your BEA WebLogic server 8.1.3 installation includes Java 1.4, which is required for this integration.
2. On an HP-UX 11i machine, ensure that the following Java-related patches are installed:

PHCO_26060	PHCO_26111	PHCO_27731	PHCO_28425
PHCO_29633	PHCO_29959	PHKL_18543	PHKL_20228
PHKL_23226	PHKL_23409	PHKL_24064	PHKL_26008
PHKL_27207	PHKL_27282	PHKL_28488	PHKL_28766
PHKL_29434	PHKL_30073	PHKL_30190	PHNE_23003
PHNE_29473	PHSS_17535	PHSS_24303	PHSS_26972
PHSS_26974	PHSS_26976	PHSS_28879	PHSS_29369
PHSS_29744	PHSS_30010	PHSS_30048	PHSS_30260
PHSS_30500			

Detailed information about these patches is available at the following web site:

<http://www.hp.com/products1/unix/java/patches>

3. After installing the BEA SSPI package for HP-UX 11.00, you must set up a COREid realm in the WebLogic domain by running a script, which, by default, does not have execute permission. Therefore, you must manually assign permission on HP-UX 11.00, then run the `setupNetPointRealm.sh` script.

4. The NetPointIdentityAsserter provider authenticates the user through the ObSSOCookie. To facilitate this, you must set up a proxy server such as Apache with a WebGate installed on it along with a WebLogic module loader.

When the WebGate successfully authenticates the user, it generates an ObSSOCookie. Using the WebLogic module loader, Apache redirects the request to the WebLogic server with the ObSSOCookie set, at which point the NetPointIdentityAsserter is invoked.

However, the WebLogic module loader for Apache 1.3.x is not shipped with BEA WebLogic Server 8.1.3. To test this functionality, you need a proxy server on a platform, such as Solaris. This functionality of the NetPoint SSPI has been tested against a proxy server on Solaris. It involves the following steps:

1. Set up a proxy server (Apache, for example) on a platform other than WebLogic (Solaris, for example).
2. Install a WebGate along with a WebLogic module loader on this proxy server.
3. Test user authentication and redirection to the WebLogic server.

## **About Parameter Names in the NetPointProvidersConfig.properties file**

Beginning with COREid version 7.0.3, you can manage NetPoint users and groups through the WebLogic Server Console

Some parameter names in the NetPointProvidersConfig.properties file have been changed as of COREid 7.0.3 while additional parameters have been introduced. If you are migrating or patching the SSPI connector from COREid 7.0.2 (or an earlier version) to COREid 7.0.4, ensure that all the parameters in NetPointProvidersConfig.properties file are set up as specified in Table 7, “NetPointProvidersConfig.properties,” on page 152. Also, read the section “Preparing the WebLogic Environment” on page 135.

## **Setting up Cookies and Header Attributes in SSPI**

The procedures for setting cookies and header attributes for SSPI are similar to those for a WebGate.

The cookie and header name and values are specified as return actions for policies defined in NetPoint. SSPI recognizes WL\_COOKIE and WL\_HeaderVar as return types for cookies and header attributes respectively.

Note that attributes and not actual headers are set in the http request. The attributes set in the request can be extracted by the target application using the `HttpServletRequest.getAttribute ()` method. To ensure that these attributes can be extracted, make sure that the target application has access to the J2EE interface `HttpServletRequest`.

## Tips

The following list is organized alphabetically according to the title in bold and are useful to understand. See also “WebLogic Portal Admin Console Changes” on page 167.

**changeUserPassword Method**—The old password will not be checked as all IDXML calls are made with admin credentials. In any case, the `changePassword` method is not called by the Portal Admin Console. The Portal Admin Console uses `resetPassword` method, which has been implemented.

**Character Restrictions for User Names**—The WebLogic Server Console imposes certain restrictions on characters allowed in user names. While all users are displayed correctly in the Console, user creation/user to role mapping must bind with these restrictions.

**Escape Mechanism for BEA WebLogic Server Role Mapping Provider**—The BEA WebLogic Server Role Mapping provider is used for defining roles in weblogic. Using the Role Mapping provider, users/groups can be assigned to roles. See “References” on page 169 for more details about WebLogic Security Roles.

This role mapper *cannot* handle various special characters in user/group names, even though LDAP allows these characters. To overcome this limitation, the NetPoint connector implements an escape mechanism that allows you to map any NetPoint group to WebLogic roles. Use Table 9 as a reference to read NetPoint groups in WebLogic Server/Portal Console.

**Table 9** Read NetPoint Groups in WebLogic Server/Portal Console

Special Character	Escaped to Sequence
:	:A
(space)	:B
, (comma)	:C
-	:D
=	:E
?	:F

**Table 9** Read NetPoint Groups in WebLogic Server/Portal Console

Special Character	Escaped to Sequence
>	:G
#	:H
' (single inverted comma)	:I
" (double inverted comma)	:J
*	:K
<	:L
&	:M
~	:N
(	:O
)	:P
{	:Q
}	:R
\t	:S
%	:T
;	:U
	:V
\$	:W

**Group Description**—Group description will *not* be available for display immediately after creating a group.

**Group Members**—Only immediate members of the group can be removed from the group. Only User members will be listed as members of group. This holds for nested groups too.

**Group With Latin Characters**—In the WebLogic Server Console, when trying to assign a group with “Latin Characters” as a parent group of another user/group, the operation fails because this interface in the console does not handle “Latin Characters”.

**Group Within a Group**—In the WebLogic Portal Console, if you try to create a group inside another group, and the new group is created, it is *not* added to the parent group. You can add the created group to the desired group using the “Add group to Group” interface in the Console.

**User/Group Workflows**—In create user/group workflows only one value for each attribute (parameter) should be specified in the configuration file.

**User/Group Display**—Unique userid (uid) will be used for all display operations for users. Group DN will be used for all display operations for groups.

**Wild Card Searches**—These searches are supported for users and groups; the match is *not* case sensitive:

- \* : returns all user/groups
- Foo\* : returns all users/groups whose name starts with Foo
- \*Foo : returns all users/groups whose name ends with Foo
- \*Foo\* : returns all users/groups whose name contains Foo
- FooBar : only returns a user/group named “FooBar”

---

**Note:** For users, a unique user identifier is matched against the pattern. However, for groups a common name is matched. For exact group searches, specify the entire group DN without escape characters.

---

See also “WebLogic Portal Admin Console Changes” on page 167.

## WebLogic Portal Admin Console Changes

The following WebLogic Portal Admin Console changes have occurred.

**Building a Group Hierarchy Tree for the NetPoint Authentication Provider**—When trying to search for groups in the WebLogic Portal Admin Console, a message may be displayed saying:

The Authentication Provider, NetPointAuthenticator, has not been configured for GUI tree mode

In this case, only one group can be searched at a time. You can use the following procedure to list all the groups in the Admin Console.

### To enable the listing of all groups in the Admin Console

1. From the WebLogic Administration Portal main menu, select Service Administration.
2. In the left pane, select Authentication Hierarchy Service.

3. In the right pane, Provider to Add to Build List field, enter the name of NetPoint authentication provider exactly (including case sensitivity):

For example:

*NetPointAuthenticator*

---

**Note:** You can find the name of NetPoint Authentication provider by selecting the Security Providers tool and expanding the Authentication Providers node.

---

4. Click Update & Build Tree.

Now, group hierarchies will be displayed for all operations where groups are involved.

**Delegated Administration**—In order for delegated administration roles to take effect, the PortalSystemDelegator role needs to be associated with the NetPoint group that was associated with the Admin role in the WebLogic Server Console during product setup.

## Configuring Multiple Policy Domains in NetPoint for Different WebLogic Servers

Using the SSPI, you can configure and apply a different NetPoint policy domain to each of several WebLogic servers.

SSPI maintains the following two configuration files in the WebLogic server domain directory:

```
NetPointProvidersConfig.properties
NetPointResourceMap.conf
```

The parameters in these two files facilitate setup of different policy domains on different WebLogic servers. Thus, all the weblogic servers in an organization need not use the same policy domain.

The SSPI integration with BEA Application Server performs three main tasks: Authentication, Role Mapping, and Authorization.

For authentication, SSPI uses the `OB_AuthnSchemeResourceName` parameter in `NetPointProvidersConfig.properties`. You define a policy in NetPoint to protect the resource specified by this parameter. Specifying different names for this authentication parameter facilitates different authentication policies for different policy domains. By default, this parameter is `/Authn/Basic`.

The same principle applies to the policy defined for anonymous authentication through `ObAuthentication.Anonymous.ResourceName`. The default value for this parameter is `/Authn/Anonymous`.

Also, different role mapping policies can be defined in different policy domains through the role mapping parameter `ObRoles.ResourceName`, which resides in *NetPointProvidersConfig.properties*. By default, this parameter is `/Authen/Roles`.

For authorization, *NetPointResourceMap.conf* contains information about mapping WebLogic resources to COREid resources. For example, a WebLogic resource of type `url` maps to a COREid resource of type `wl_url`. To make different WebLogic servers refer to resource protection policies defined in different domains, specify different resource mappings for your various WebLogic servers.

For example, one WebLogic server can map `url` to `wl_wrl` for HTTP resources and protect resources of type `wl_url` in its policy domain. A second weblogic server can map `url` to `weblogic_wrl` for HTTP resources and protect resources of type `weblogic_url` in its policy domain.

---

**Note:** For a given WebLogic server, the authentication and role mapping policies are picked up from a single policy domain, whereas for authorization, the policies used to protect resources are picked up from any domain that has a policy defined for the resource type being accessed.

---

## References

The following alphabetical list includes links to WebLogic documents that discuss concepts such as WebLogic security, resources, and roles.

- **EJB Security-related Deployment Descriptors**—[http://e-docs.bea.com/wls/docs81/security/ejb\\_client.html #1033936](http://e-docs.bea.com/wls/docs81/security/ejb_client.html#1033936).
- **Introduction to WebLogic Security**—<http://e-docs.bea.com/wls/docs81/secintro>.
- **Security Framework Whitepaper**—[http://www.bea.com/content/news\\_events/white\\_papers/BEA\\_WL\\_Server\\_TechSecurity\\_wp.pdf](http://www.bea.com/content/news_events/white_papers/BEA_WL_Server_TechSecurity_wp.pdf).
- **Terminology for the Newly Introduced Concepts**—<http://e-docs.bea.com/wls/docs81/secintro/terms.html>.
- **Types of WebLogic Resources**—<http://e-docs.bea.com/wls/docs81/secwres/types.html>.
- **Webapp security related deployment descriptors**—[http://e-docs.bea.com/wls/docs81/security/thin\\_client.html #1045984](http://e-docs.bea.com/wls/docs81/security/thin_client.html#1045984).
- **WebLogic Security Roles**—<http://e-docs.bea.com/wls/docs81/secwres/secroles.html>

# Troubleshooting NetPoint Security Provider for WebLogic

**Problem**—I get the following error when starting the WebLogic Server:

```
<May 23, 2003 1:44:07 AM PDT> <Error> <OblixSecurityProviders>  
<700021> <Authentication failed for user [admin]. Reason -  
Unprotected resource LOGIN Authen:/Authen/Basic used in an  
ObAuthenticationScheme or ObuserSession constructor.>
```

**Solution**—The authentication policy may be disabled or absent. See “Setting Up WebLogic Policies in NetPoint” on page 117 for setup information.

**Problem**—I created a policy to protect web application (for example, /security) with an authorization rule to allow access to some users. But, in reality everyone is allowed access.

**Solution**—Ensure the following:

- Enable the policy.
- Do not use host identifiers.
- Use the Access Tester to ensure that the policy is being evaluated for this resource and the user who is allowed to access the resource.

Note that the default behavior of Netpoint Access Server is to allow access if the resource is not protected. This default behavior can be changed by modifying the “denyOnProtect” flag in the Access Server configuration.

**Problem**— I want to use the methods of Oblix principal ObWLSUser object in an EJB.

**Solution**—You can use all the methods exposed by the Principal object, but not methods like “isAuthorized” that are present only in ObWLSUser object. This is because they make JNI calls to Access server SDK, which is not present on the client side except when the EJB is being run from another WebLogic Server using Oblix Providers. If you really want to execute methods like “isAuthorized” on the client side, then you must install Access SDK and call ObConfig.Init() before making any calls. A suggested solution is to get the session token from the ObWLSUser object and then use Access SDK methods to execute methods such as isAuthorized.

**Problem**—The server does not start up. The log has the following entry

```
####<May 29, 2003 2:07:48 PM PDT> <OFF> <Unknown> <vjain>  
<exampleserver> <main> <<WLS Kernel>> <> <000000> <Message text not  
found - Can't locate bundle for class>
```

**Solution**—Netpoint Security Providers message resources are bundled in `wlNetpoint.jar`. Make sure that `wlNetpoint.jar` is present in the classpath. Refer to “Preparing the Environment” on page 112 for details on setting up the required environment variables.

**Problem**—I get the following error:

```
<May 29, 2003 2:36:40 PM PDT> <Error> <OblixSecurityProviders>
<700053> <Exception encountered when isProtected() called for the
resource - Type=wl_svr, isEnabled=true, URL=/exampleserver,
operation=default. Reason - The requested resource could not be
mapped to a policy domain in the Policy database. Check if the
corresponding directory service is up.>
```

**Solution**—The `wl_svr` resource type is probably enabled in `NetpointResourceMap.txt` but the resource type is not yet created in the Netpoint Access System.

**Problem**—The installer does not write the complete file and the code expects some default values.

**Solution**—This problem occurs when some of the values are not specified in the properties file. Use the file in `install_dir\weblogic8\examples` and modify it for your environment.

For example, if an action is not specified for `authen` to get name later:

```
,Roles:{Regular=Regular} ,Resource:type=<url>,
application=security, contextPath=/security, uri=/admin/edit.jsp,
httpMethod=GET ,ContextHandler:HttpServletRequest
,ContextHandler:HttpServletResponse>
```

```
<Jun 10, 2003 3:37:21 PM PDT> <Debug> <OblixSecurityProviders>
<000000> <OblixResource got from cache>
```

```
<Jun 10, 2003 3:37:21 PM PDT> <Debug> <OblixSecurityProviders>
<000000> <Entering OblixDatabase.isProtected for resource
Type=wl_url, isEnabled=true, URL=/security/admin/edit.jsp,
operation=GET>
```

```
<Jun 10, 2003 3:37:21 PM PDT> <Debug> <OblixSecurityProviders>
<000000> <Is resource protected? true>
```

```
<Jun 10, 2003 3:37:21 PM PDT> <Debug> <OblixSecurityProviders>
<000000> <Entering OblixDatabase.isAccessAllowed for cn=Drusy
Sails,ou=LHuman Resource,ou=Los Angles,ou=Dealer1k1,ou=Latin
America,ou=Ford,o=Company,c=US on resource Type=wl_url,
isEnabled=true, URL=/security/admin/edit.jsp, operation=GET>
```

```
<Jun 10, 2003 3:37:21 PM PDT> <Debug> <OblixSecurityProviders>
<000000> <OblixDatabase.isAccessAllowed returned PERMIT>
```

```
<Jun 10, 2003 3:37:21 PM PDT> <Debug> <OblixSecurityProviders>
<000000> <Found an authenticated principal of type ObWLSUser>
```

```
<Jun 10, 2003 3:37:21 PM PDT> <Debug> <OblixSecurityProviders>
<000000> <Got a SSO token>
```

Rest of stack:

```

<Jun 10, 2003 3:37:21 PM PDT> <Debug> <OblixSecurityProviders> <000000> <cookie domain = null>
<Jun 10, 2003 3:37:59 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Entering ConfigurationUpdate.run>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Inside Login.jsp>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Calling ServletAuthentication.authenticate>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Entering
OblixAuthenticationProviderImpl.getLoginModuleConfiguration>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Entering
OblixAuthenticationProviderImpl.getConfiguration>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Entering OblixLoginModuleImpl.initialize>
<Jun 10, 2003 3:38:20 PM PDT> <Info> <OblixSecurityProviders> <700006> <OblixLoginModule called in normal mode.>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Entering OblixLoginModuleImpl.Login>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Entering OblixLoginModuleImpl.getCallbacks>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <userName= [obdummyuser]>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Form login with ObSSOCookie already set
[yGNpe11Seo0FNGGB6LMmwzSEVG3Mz9Gdzy84bw0SqjEFYr7zGF1RpjklQxwTmtYaneQVdtzRMLsrsRwXlqshh3mObyIn82woISlyA8GVzbzBzRpZLTJ0uh
wynFKjd2Bj8ZTNxh1lqqadRBvIJ0wOYsbtXoIE%2Bg1DY27F%2B5nAm0ZF18fh5Ae1wFERq6ktUX6sw6DKRkn%2FRAoujkZydp4%2Bt55vqpzkSG1IECKe
Le12Nxp58qDBX0UL3vsj8na1w0ts0N%2FjiQrklTq6HGTGI1vYeN5%2B81rJmqSb1S22cHx3PYzvy1Kb9R9tb5toMQHi a3tz53wvKudaUblJEwvt93mbJMO
PPVbBzpkIQDPgi]>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Entering OblixDatabase.getPrincipal with
ObSSOCookie =
[yGNpe11Seo0FNGGB6LMmwzSEVG3Mz9Gdzy84bw0SqjEFYr7zGF1RpjklQxwTmtYaneQVdtzRMLsrsRwXlqshh3mObyIn82woISlyA8GVzbzBzRpZLTJ0uh
wynFKjd2Bj8ZTNxh1lqqadRBvIJ0wOYsbtXoIE%2Bg1DY27F%2B5nAm0ZF18fh5Ae1wFERq6ktUX6sw6DKRkn%2FRAoujkZydp4%2Bt55vqpzkSG1IECKe
Le12Nxp58qDBX0UL3vsj8na1w0ts0N%2FjiQrklTq6HGTGI1vYeN5%2B81rJmqSb1S22cHx3PYzvy1Kb9R9tb5toMQHi a3tz53wvKudaUblJEwvt93mbJMO
PPVbBzpkIQDPgi]>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <ObUserSession status = [1]>
<Jun 10, 2003 3:38:20 PM PDT> <Info> <OblixSecurityProviders> <700014> <ObSSOCookie status [1]. 1 - Logged in, 2 - Logged
out, 4 - Session expired.>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Entering OblixLoginModuleImpl.abort>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <ServletAuthentication.authenticate result = 1>
<Jun 10, 2003 3:38:20 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Invalid ObSSOCookie - cannot do SSO.>

```

**Solution**—Set values for the parameters as shown below in the `NetPointProvidersConfig.properties` file:

```

ObAuthenticationScheme.AuthorizationRule.ActionType=WL_REALM
ObAuthenticationScheme.AuthorizationRule.ActionName=uid

```

**Problem**—I get the following stack.

```

<Jun 9, 2003 7:43:04 PM PDT> <Debug> <OblixSecurityProviders>
<000000> <userName= [weblogic_system]>
<Jun 9, 2003 7:43:04 PM PDT> <Debug> <OblixSecurityProviders>
<000000> <Entering OblixDatabase.login() for user weblogic_system>
<Jun 9, 2003 7:43:04 PM PDT> <Debug> <OblixSecurityProviders>
<000000> <Authentication failed for [weblogic_system] with message
Unprotected resource LOGIN Authen:/Authen/Basic8 used in an
ObAuthenticationScheme or ObUserSession constructor.
com.oblix.access.ObAccessException: Unprotected resource LOGIN
Authen:/Authen/Basic8 used in an ObAuthenticationScheme or
ObUserSession constructor.
at com.oblix.access.ObUserSession.initCppSideAuthenticate(Native
Method)
at com.oblix.access.ObUserSession.<init>(ObUserSession.java:222)
at com.oblix.weblogic.internal.OblixDatabase.login
(OblixDatabase.java:185)
Rest of stack:

```

```

at com.oblix.weblogic.security.providers.authentication.OblixLoginModuleImpl.login(OblixLoginModuleImpl.java:161)
at weblogic.security.service.DelegateLoginModuleImpl.login(DelegateLoginModuleImpl.java:71)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:324)
at javax.security.auth.login.LoginContext.invoke(LoginContext.java:675)
at javax.security.auth.login.LoginContext.access$000(LoginContext.java:129)
at javax.security.auth.login.LoginContext$4.run(LoginContext.java:610)

```

```

at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.login.LoginContext.invokeModule(LoginContext.java:607)
at javax.security.auth.login.LoginContext.login(LoginContext.java:534)
at weblogic.security.service.PrincipalAuthenticator.authInternal(PrincipalAuthenticator.java:329)
at weblogic.security.service.PrincipalAuthenticator.authenticate(PrincipalAuthenticator.java:282)
at weblogic.security.service.SecurityServiceManager.doBootAuthorization(SecurityServiceManager.java:913)
at weblogic.security.service.SecurityServiceManager.initialize(SecurityServiceManager.java:1036)
at weblogic.t3.srvr.T3Srvr.initializeHere(T3Srvr.java:783)
at weblogic.t3.srvr.T3Srvr.initialize(T3Srvr.java:627)
at weblogic.t3.srvr.T3Srvr.run(T3Srvr.java:337)
at weblogic.Server.main(Server.java:32)
>
<Jun 9, 2003 7:43:04 PM PDT> <Error> <OblixSecurityProviders> <700021> <Authentication failed for user [weblogic_system]. Reason - Unprotected resource LOGIN Authen:/Authen/Basic8 used in an ObAuthenticationScheme or ObUserSession constructor.>
<Jun 9, 2003 7:43:04 PM PDT> <Debug> <OblixSecurityProviders> <000000> <ObUserSession ctor from username, password failed com.oblix.access.ObAccessException: Unprotected resource LOGIN Authen:/Authen/Basic8 used in an ObAuthenticationScheme or ObUserSession constructor.
at com.oblix.access.ObUserSession.initCppSideAuthenticate(Native Method)
at com.oblix.access.ObUserSession.<init>(ObUserSession.java:222)
at com.oblix.weblogic.internal.oblixDatabase.login(OblixDatabase.java:185)
at com.oblix.weblogic.security.providers.authentication.OblixLoginModuleImpl.login(OblixLoginModuleImpl.java:161)
at weblogic.security.service.DelegateLoginModuleImpl.login(DelegateLoginModuleImpl.java:71)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:324)
at javax.security.auth.login.LoginContext.invoke(LoginContext.java:675)
at javax.security.auth.login.LoginContext.access$000(LoginContext.java:129)
at javax.security.auth.login.LoginContext$4.run(LoginContext.java:610)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.login.LoginContext.invokeModule(LoginContext.java:607)
at javax.security.auth.login.LoginContext.login(LoginContext.java:534)
at weblogic.security.service.PrincipalAuthenticator.authInternal(PrincipalAuthenticator.java:329)
at weblogic.security.service.PrincipalAuthenticator.authenticate(PrincipalAuthenticator.java:282)
at weblogic.security.service.SecurityServiceManager.doBootAuthorization(SecurityServiceManager.java:913)
at weblogic.security.service.SecurityServiceManager.initialize(SecurityServiceManager.java:1036)
at weblogic.t3.srvr.T3Srvr.initializeHere(T3Srvr.java:783)
at weblogic.t3.srvr.T3Srvr.initialize(T3Srvr.java:627)
at weblogic.t3.srvr.T3Srvr.run(T3Srvr.java:337)
at weblogic.Server.main(Server.java:32)
>
<Jun 9, 2003 7:43:04 PM PDT> <Debug> <OblixSecurityProviders> <000000> <Entering OblixLoginModuleImpl.abort>
<Jun 9, 2003 7:43:04 PM PDT> <Critical> <Security> <BEA-090402> <Authentication denied: Boot identity not valid; The user name and/or password from the boot identity file (boot.properties) is not valid. The boot identity may have been changed since the boot identity file was created. Please edit and update the boot identity file with the proper values of username and password. The first time the updated boot identity file is used to start the server, these new values are encrypted.>
*****
The webLogic Server did not start up properly.
Reason: weblogic.security.SecurityInitializationException: Authentication denied: Boot identity not valid; The user name and/or password from the boot identity file (boot.properties) is not valid. The boot identity may have been changed since the boot identity file was created. Please edit and update the boot identity file with the proper values of username and password. The first time the updated boot identity file is used to start the server, these new values are encrypted.
*****

```

**Solution**—Check the resource, operation and policy and authorization rules. Some resource is not protected.

**Problem**—I get the following stack.

```

4549, name=exampleswebApp, context-path=/exampleswebApp) * .jsp:
param compilerval
  initialized to: com.sun.tools.javac.Main>
<Jun 9, 2003 7:23:53 PM PDT> <Info> <HTTP> <BEA-101047>
<[ServletContext(id=2614
4549, name=exampleswebApp, context-path=/exampleswebApp) * .jsp:
param pageCheckSe
conds initialized to: 1>
<Jun 9, 2003 7:23:53 PM PDT> <Info> <HTTP> <BEA-101047>
<[ServletContext(id=2614
4549, name=exampleswebApp, context-path=/exampleswebApp) * .jsp:
param encoding in

```

```

ialized to: null>
<Jun 9, 2003 7:23:53 PM PDT> <Info> <HTTP> <BEA-101047>
<[ServletContext(id=2614
4549,name=exampleswebApp,context-path=/exampleswebApp)] *.jsp:
param superClass
Rest of stack:

```

---

```

initialized to weblogic.servlet.jsp.JspBase>
<Jun 9, 2003 7:23:53 PM PDT> <Info> <HTTP> <BEA-101047> <[ServletContext(id=2614
4549,name=exampleswebApp,context-path=/exampleswebApp)] *.jsp: param srcCompiler
initialized to weblogic.jspc>
<Jun 9, 2003 7:23:53 PM PDT> <Info> <HTTP> <BEA-101047> <[ServletContext(id=2614
4549,name=exampleswebApp,context-path=/exampleswebApp)] *.jsp: param workingDir
initialized to: C:\bea\user_projects\examplesNP\examplesServer\wlnotdelete\extr
act\examplesServer_exampleswebApp_exampleswebApp>
<Jun 9, 2003 7:23:53 PM PDT> <Info> <HTTP> <BEA-101047> <[ServletContext(id=2614
4549,name=exampleswebApp,context-path=/exampleswebApp)] *.jsp: initialization complete>
<Jun 9, 2003 7:23:57 PM PDT> <Info> <HTTP> <BEA-101047> <[ServletContext(id=2614
4549,name=exampleswebApp,context-path=/exampleswebApp)] /*: init>
<Jun 9, 2003 7:23:57 PM PDT> <Info> <HTTP> <BEA-101047> <[ServletContext(id=2614
4549,name=exampleswebApp,context-path=/exampleswebApp)] /*: Using standard I/O>
<Jun 9, 2003 7:24:07 PM PDT> <Info> <HTTP> <BEA-101047> <[ServletContext(id=2337
0522,name=console,context-path=/console)] actions: init>
<Jun 9, 2003 7:24:21 PM PDT> <Info> <Management> <BEA-140009> <Configuration cha
nges for the domain have been saved to the repository.>
<Jun 9, 2003 7:24:45 PM PDT> <Warning> <RMI> <BEA-080003> <RuntimeException thro
wn by rmi server: weblogic.management.internal.RemoteMBeanServerImpl.invoke(Ljav
ax.management.ObjectName;Ljava.lang.String;[Ljava.lang.Object;[Ljava.lang.String
;)>
  javax.management.RuntimeOperationsException: RuntimeException thrown by the inv
oke method of the Dynamic MBean.
  java.lang.ClassCastException: java.lang.NoClassDefFoundError
  at javax.management.modelmbean.RequiredModelMBean.invoke (RequiredModelMBean.java:1166)
  at com.sun.management.jmx.MBeanServerImpl.invoke (MBeanServerImpl.java:1557)
  at com.sun.management.jmx.MBeanServerImpl.invoke (MBeanServerImpl.java:1525)
  at weblogic.management.internal.RemoteMBeanServerImpl.invoke (RemoteMBeanServerImpl.java:763)
  at weblogic.management.internal.RemoteMBeanServerImpl_WLSkel.invoke(Unknown Source)
  at weblogic.rmi.internal.BasicServerRef.invoke (BasicServerRef.java:407)
  at weblogic.rmi.internal.BasicServerRef$1.run (BasicServerRef.java:356)
  at weblogic.security.acl.internal.AuthenticatedSubject.doAs (AuthenticatedSubject.java:353)
  at weblogic.security.service.SecurityManager.runAs (SecurityManager.java:123)
  at weblogic.rmi.internal.BasicServerRef.handleRequest (BasicServerRef.java:351)
  at weblogic.rmi.internal.BasicExecuteRequest.execute (BasicExecuteRequest.java:30)
  at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:178)
  at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:151)
----- nested within: -----
  javax.management.RuntimeOperationsException: RuntimeException thrown by the invoke method of the Dynamic MBean
  at com.sun.management.jmx.MBeanServerImpl.invoke (MBeanServerImpl.java:1559)
  at com.sun.management.jmx.MBeanServerImpl.invoke (MBeanServerImpl.java:1525)
  at weblogic.management.internal.RemoteMBeanServerImpl.invoke (RemoteMBeanServerImpl.java:763)
  at weblogic.management.internal.RemoteMBeanServerImpl_WLSkel.invoke(Unknown Source)
  at weblogic.rmi.internal.BasicServerRef.invoke (BasicServerRef.java:407)
  at weblogic.rmi.internal.BasicServerRef$1.run (BasicServerRef.java:356)
  at weblogic.security.acl.internal.AuthenticatedSubject.doAs (AuthenticatedSubject.java:353)
  at weblogic.security.service.SecurityManager.runAs (SecurityManager.java:123)
  at weblogic.rmi.internal.BasicServerRef.handleRequest (BasicServerRef.java:351)
  at weblogic.rmi.internal.BasicExecuteRequest.execute (BasicExecuteRequest.java:30)
  at weblogic.kernel.ExecuteThread.execute(ExecuteThread.java:178)
  at weblogic.kernel.ExecuteThread.run(ExecuteThread.java:151)
>
<Jun 9, 2003 7:24:46 PM PDT> <Info> <Management> <BEA-140009> <Configuration changes for the domain have been saved to
the repository.>
<Jun 9, 2003 7:25:14 PM PDT> <Info> <HTTP> <BEA-101047> <[ServletContext(id=23370522,name=console,context-path=/console)]
FileServlet: init>
<Jun 9, 2003 7:25:28 PM PDT> <Info> <HTTP> <BEA-101047> <[ServletContext(id=2337
0522,name=console,context-path=/console)] FileServlet: Using standard I/O>

```

---

**Solution**—You are probably missing the NetPoint configuration file in the WLS Domain directory.

**Problem**—I get the following stack.

```
<Jun 9, 2003 6:11:39 PM PDT> <Info> <WebLogicServer> <BEA-000377>
<Starting WebLogic Server with Java HotSpot(TM) Client VM Version
1.4.1_02-ea-b01 from Sun Microsystems Inc.>
```

```
<Jun 9, 2003 6:11:41 PM PDT> <Info> <Configuration Management>
<BEA-150016> <This server is being started as the administration
server.>
```

```
<Jun 9, 2003 6:11:41 PM PDT> <Info> <Management> <BEA-141107>
<Version: WebLogic Server 8.1 Thu Mar 20 23:06:05 PST 2003 246620
WebLogic XMLX Module 8.1 Thu Mar 20 23:06:05 PST 2003 246620 >
```

```
<Jun 9, 2003 6:11:43 PM PDT> <Notice> <Management> <BEA-140005>
<Loading domain configuration from configuration repository at /
export/home/BEA81/user_projects/mydomain/./config.xml.>
```

```
<Jun 9, 2003 6:11:57 PM PDT> <Info> <Logging> <000000> <FileLogger
Opened at ./myserver/myserver.log>
```

```
<Jun 9, 2003 6:12:03 PM PDT> <Error> <Unknown> <000000> <Unable to
access undefined message, id=700025>
```

```
*****
```

```
The WebLogic Server did not start up properly.
```

```
java.lang.ExceptionInInitializerError
```

```
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native
Method)
```

```
at sun.reflect.NativeConstructorAccessorImpl.newInstance
(NativeConstructorAccessorImpl.java:39)
```

```
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance
(DelegatingConstructorAccessorImpl.java:27)
```

```
at java.lang.reflect.Constructor.newInstance(Constructor.java:274)
```

```
at java.lang.Class.newInstance0(Class.java:306)
```

```
at java.lang.Class.newInstance(Class.java:259)
```

```
at weblogic.security.service.SecurityServiceManager.
createSecurityProvider (SecurityServiceManager.java:1686)
```

```
at weblogic.security.service.RoleManager.initialize
(RoleManager.java:147)
```

```
at weblogic.security.service.RoleManager.<init>
(RoleManager.java:93)
```

```
at weblogic.security.service.SecurityServiceManager.doRole
(SecurityServiceManager.java:1417)
```

```
at weblogic.security.service.SecurityServiceManager.
initializeRealm (SecurityServiceManager.java:1271)
```

```
at weblogic.security.service.SecurityServiceManager.loadRealm
(SecurityServiceManager.java:1216)
```

```
at weblogic.security.service.SecurityServiceManager.
initializeRealms (SecurityServiceManager.java:1338)
```

```
at weblogic.security.service.SecurityServiceManager.initialize
(SecurityServiceManager.java:1018)
```

```
at weblogic.t3.srvr.T3Srvr.initializeHere(T3Srvr.java:783)
```

```
at weblogic.t3.srvr.T3Srvr.initialize(T3Srvr.java:627)
```

```

at weblogic.t3.srvr.T3Srvr.run(T3Srvr.java:337)
at weblogic.Server.main(Server.java:32)
Caused by:
com.oblix.weblogic.configuration.ConfigurationException: Unable to
access undefined message, id=700025
at com.oblix.weblogic.configuration.OblixConfiguration.
loadProperties (OblixConfiguration.java:106)
at com.oblix.weblogic.configuration.OblixConfiguration.<init>
(OblixConfiguration.java:61)
at com.oblix.weblogic.configuration.OblixConfiguration.getInstance
(OblixConfiguration.java:55)
at com.oblix.weblogic.internal.OblixDatabase.<init>
(OblixDatabase.java:47)
at com.oblix.weblogic.internal.OblixDatabase.getInstance
(OblixDatabase.java:39)
at com.oblix.weblogic.security.providers.roles.
OblixRoleMapperProviderImpl.<clinit>(OblixRoleMapperProviderImpl.j
ava:51)
... 18 more
*****

```

**Solution**—The Oblix Jar file is not correctly packaged.

## Additional Resources

The following URLs are provided for background:

- Security framework whitepaper:  
[http://www.bea.com/content/news\\_events/white\\_papers/BEA\\_WL\\_Server\\_TechSecurity\\_wp.pdf](http://www.bea.com/content/news_events/white_papers/BEA_WL_Server_TechSecurity_wp.pdf)
- Types of WebLogic Resources:  
<http://e-docs.bea.com/wls/docs81/secwres/types.html>
- Roles:  
<http://e-docs.bea.com/wls/docs81/secwres/secroles.html>
- Web application security deployment descriptors:  
[http://e-docs.bea.com/wls/docs81/security/thin\\_client.html#1045984](http://e-docs.bea.com/wls/docs81/security/thin_client.html#1045984)
- EJB security deployment descriptors:  
[http://e-docs.bea.com/wls/docs81/security/ejb\\_client.html#1033936](http://e-docs.bea.com/wls/docs81/security/ejb_client.html#1033936)
- Descriptions of terms and concepts:  
<http://e-docs.bea.com/wls/docs81/secintro/terms.html>





# 4 Integrating NetPoint with IBM WebSphere

The NetPoint Connector for WebSphere enables you to integrate applications running on IBM's WebSphere Application Server with NetPoint authentication and authorization services. The connector also makes NetPoint-managed users and groups available for authentication and authorization within WebSphere.

This chapter describes how prepare your environment, then install, set up, and test the NetPoint Connector for WebSphere, and configure your WebSphere Application Server for NetPoint. This chapter covers the following topics:

- “About the NetPoint Connector for WebSphere” on page 180
- “Integration Architecture” on page 184
- “Supported Versions and Platforms” on page 189
- “Preparing to Install the NetPoint Connector” on page 190
- “Installing the NetPoint Connector for WebSphere” on page 206
- “Completing NetPoint Connector Setup” on page 217
- “Configuring WebSphere Application Server v4” on page 221
- “Integrating NetPoint with the Portal v4” on page 235
- “Configuring WebSphere Application Server v5” on page 247
- “Integrating NetPoint and WebSphere Portal v5” on page 255
- “Configuring the WebSphere Application Server v6” on page 276
- “Configuration Files” on page 283
- “Implementation Notes for the TAI” on page 292
- “Implementation Notes for Active Directory” on page 294
- “Troubleshooting the NetPoint Connector for WebSphere” on page 295

# About the NetPoint Connector for WebSphere

The NetPoint Connector for WebSphere enables WebSphere Application Server administrators to integrate applications running on WebSphere with NetPoint authentication and authorization services.

Using the NetPoint Connector for WebSphere, users who try to access NetPoint-protected WebSphere resources are challenged and authenticated by the NetPoint Access System. The connector also makes NetPoint-managed users and groups available for authentication and authorization.

Advantages of using the NetPoint Connector for WebSphere include:

- Providing information about NetPoint-managed users and groups to the WebSphere Security Server for authentication and authorization.
- Authenticating users who access WebSphere resources such as JSPs, EJBs, and Servlets.

You can use NetPoint authentication schemes to provide single sign-on for Web and non-Web enabled applications.

- Authorizing users who access WebSphere resources.

NetPoint supports WebSphere's security role-based authorization. The WebSphere's role-based authorization grants access to a protected resource based on a role (such as Manager) for a user or group. NetPoint integrates WebSphere Server security roles with NetPoint policy-based authorization.

- Protecting WebSphere Server resources such as administration tools, Events, servlets, passwords, JDBC connection pools, JMS destinations, and JNDI contexts.

You can use WebSphere's role-based access policies to control access to WebSphere resources.

- Enabling single sign-on between a NetPoint-protected resource and a WebSphere Server resource that is protected with WebSphere Security constraints.

Integrating NetPoint with the WebSphere Application Server includes several procedures, as introduced below.

## Task overview: Integrating NetPoint with the WebSphere Application Server

1. Prepare your environment before you begin to install the NetPoint Connector, as described in “Preparing Your Environment” on page 191.
2. Install the NetPoint Connector, as described in “Installing the NetPoint Connector for WebSphere” on page 206.
3. Complete setup for, and test, the NetPoint Connector, as described in “Completing NetPoint Connector Setup” on page 217.
4. Configure your WebSphere Application Server, as described in:
  - “Configuring WebSphere Application Server v5” on page 247
  - “Configuring WebSphere Application Server v4” on page 221
5. Integrate NetPoint with the WebSphere Portal Server, if this is part of your environment, as described in:
  - “Integrating NetPoint and WebSphere Portal v5” on page 255
  - “Integrating NetPoint with the Portal v4” on page 235

As you complete activities in this chapter, you will see the following path name formats:

***COREid\_install\_dir***—The directory where you installed the NetPoint COREid Server. In your installation, for example, this may be C:\NetPoint. During installation the \identity subdirectory is added to the specified destination making the full path to the directory where you installed the COREid Server *COREid\_install\_dir\identity*.

***WebGate\_install\_dir***—The directory where you installed the NetPoint WebGate. In your installation, for example, this may be C:\NetPoint\Webcomponent. During installation the \access subdirectory is added to the specified destination making the full path to the directory where you installed the WebGate *WebGate\_install\_dir\access*.

***NPCWS\_install\_dir***—The directory where the NetPoint Connector for WebSphere was installed. In your installation, for example, this may be C:\NetPoint\NetPointWASRegistry.

***WAS\_install\_dir***—The directory where the WebSphere Application Server is installed. In your installation, for example, this may be C:\IBM\WebSphere\AppServer.

***WPS\_install\_dir***—The directory where the WebSphere Portal Server is installed. In your installation, for example, this may be C:\IBM\WebSphere\PortalServer. This directory is used for the Portal logs such as, *WPS\_install\_dir/log/appserver-out.log*.

When you see one of the notations above, simply substitute the appropriate path name for your environment as you complete the step.

## WebSphere Components

For complete support information, see “Supported Versions and Platforms” on page 189. The following WebSphere components are used in the integration between WebSphere and NetPoint:

**WebSphere Application Server (WAS)**—The WebSphere Application Server (WAS) enables secure, high volume transactions and Web services.

- WAS 4.0.2 is J2EE 1.2 compliant.
- WAS 5.0/5.1 is J2EE 1.3 compliant
- WAS 6 is J2EE 1.4 compliant

---

**Note:** Both implement authorization using the EJB1.1 specification for security roles. A security role is a set of permissions for access to Web resources and specific EJB methods.

---

**WebSphere Portal Server (WPS)**—The WebSphere Portal Server provides single sign-on for portlets based on the Java Authentication and Authorization Services (JAAS). The single sign-on implementation allows portlet developers to extract:

- A user’s username and password, and distinguished name (DN)
- The DNs of any groups that the user belong to
- The WebSphere Application Server CORBA Credential
- The LTPA token

Any combination of these objects can be used to provide single sign-on to the portlet’s back end. For example, the username and password may be used to create a Basic Authentication HTTP header. Or the LTPA Token may be used to provide single sign-on to another WebSphere Application Server in the same security domain.

**Web Trust Association Interceptor (TAI)**—The TAI is used for third-party proxy authentications. The WebSphere Application Server uses the TAI to enforce the trust policy between WebSphere and third-party security providers for single sign-on. The TAI allows NetPoint to authenticate users who try to access resources on the WebSphere Application Server.

**Application Assembly Tool (AAT)**—The AAT is used to build security-aware applications. You use the AAT to define WebSphere roles and bind them to NetPoint users and groups.

## NetPoint Connector for WebSphere Components

The NetPoint Connector for WebSphere uses the Trust Association Interceptor (TAI) and the NetPoint COREid System and Access System components. In addition, the following NetPoint components are included with the Connector for WebSphere:

**NetPointWASRegistry**—This is the NetPoint Connector for WebSphere, which is a NetPoint implementation of the WebSphere CustomRegistry (also known as a custom user registry (CUR)). The NetPointWASRegistry serves as a plug-in to the WebSphere Application Server (WAS).

The WebSphere CustomRegistry defines the methods used by the WAS to perform security operations for applications configured to use this. For example, the WebSphere CustomRegistry may be used to identify critical attributes, such as username and password, and to combine user information from diverse data sources.

The NetPointWASRegistry consists of the Access Server SDK and Identity XML. The NetPointWASRegistry establishes a native connection between the WAS and NetPoint, enabling WebSphere customers to use NetPoint policy-based security to control user access to business applications.

**IdentityXML**—The NetPoint Connector for WebSphere uses IdentityXML calls to get user and group information from the COREid Server. Typically, you use IdentityXML to integrate NetPoint with external software systems and to perform NetPoint functions programmatically, without using a browser.

**Access Server SDK**—The Access Server Software Developer's Kit (SDK) allows you to create an interface that can be built into WebSphere and to create an AccessGate that communicates with the Access Server for authentication purposes. The Access Server SDK is automatically installed when you install NetPoint Connector for WebSphere and is used by the TAI.

**NetPoint Custom Member Repository (CMR)**—The NetPoint CMR is an extension of the NetPointWASRegistry (a custom user registry) that resides on the WebSphere Portal Server. The WebSphere Portal Server uses WebSphere Application Server security for authentication when logging in to the Portal. The WebSphere Portal Server enables users to customize and personalize their experience and uses a component called Member Services to manage information about users, user accounts, user profile attributes, and group memberships.

The NetPoint CMR is an instance of a Member Services component that connects the WebSphere Portal Server to the NetPoint COREid system users and groups. The CMR implements the IBM WebSphere MemberRepository interface, and is used to assign and determine access control to the portlets. The NetPoint CMR stores user.baseattributes and group.baseattributes and supports *only* read operations, not create or modify or delete operations.

The WebSphere Portal Server will use the NetPoint CMR to make IdentityXML queries like `getAttributes` for a user for personalization, `getGroupMemberships`, search users by attribute, and similar functions.

---

**Note:** The NetPoint 7.0.2 Connector for WebSphere does *not* support `getGroupMemberships`. As a result, in the case of Nested Groups, if you check for inner group membership the parent group details will not be displayed.

---

For more information, see the “Supported Versions and Platforms” on page 189.

## Integration Architecture

The integration between WebSphere and NetPoint can vary depending on whether you use only the NetPointWASRegistry or if you also use NetPoint single sign-on. For details, see:

- “Scenario 1: Use of NetPointWASRegistry” on page 184
- “Scenario 2: Architecture for Single Sign-On” on page 186

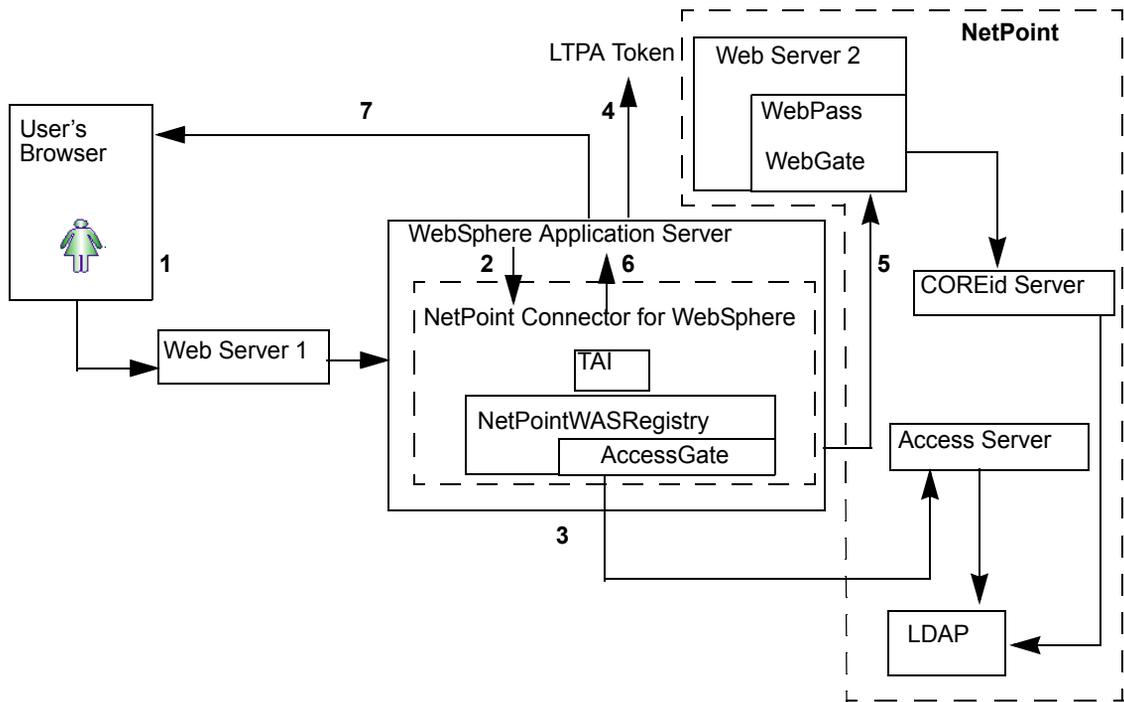
For additional information, see “Mapping NetPoint Users and Groups to Security Roles in WAS” on page 188. See also “Integration Scenario with the NetPoint CMR” on page 237.

### Scenario 1: Use of NetPointWASRegistry

The NetPointWASRegistry obtains NetPoint-managed user and group information and performs authentication based on that information.

Figure 6 illustrates an implementation of the NetPoint Connector for WebSphere using the NetPointWASRegistry.

**Figure 6** Integrating the WebSphere Application Server with the NetPointWASRegistry



In this scenario, use of a WebGate is optional. The WebGate is needed only for single sign-on or to protect the WebPass.

### **Process overview: Login using WAS with the NetPointWASRegistry**

1. A user tries to access a WebSphere resource through a browser.
2. The WAS forwards the user's request to the NetPoint Connector for WebSphere.
3. NetPoint Connector for WebSphere checks with the Access Server and authenticates the user.
4. If single sign-on is enabled in the WAS, an LTPA token is generated.
5. The NetPoint Connector for WebSphere queries the COREid Server via WebPass for a list of groups to which the user belongs.

The COREid Server checks the directory and returns the information to the NetPoint Connector for WebSphere.

6. NetPoint Connector for WebSphere returns this information to the WAS.
7. The WAS checks the deployment descriptor for a user-security or group-security role mapping. If the user or group belongs to a security role that

is allowed to access the resource, the WAS allows the user to access the resource.

---

**Note:** The WAS and both the Web servers must belong to the same domain.

---

## Scenario 2: Architecture for Single Sign-On

The NetPoint single sign-on feature allows NetPoint-authenticated users to access any NetPoint-protected resource without having to re-authenticate. To use NetPoint single sign-on, you must enable the TAI and install an AccessGate plug-in on the Web server servicing WAS.

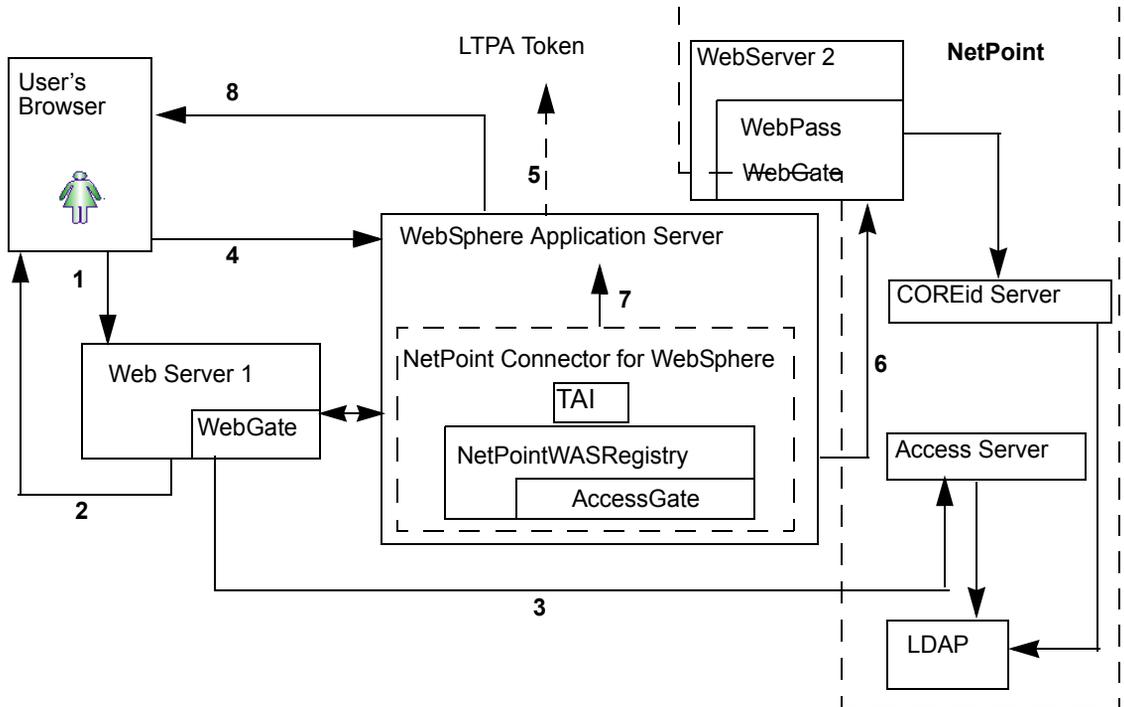
---

**Note:** A WebGate is required for single sign-on.

---

Figure 7 illustrates WAS using NetPoint single sign-on.

**Figure 7** NetPoint SSO with the WebSphere Application Server



### Process overview: Login using the WAS with NetPoint SSO

1. The user attempts to access a NetPoint-protected WebSphere resource.
2. WebGate (or AccessGate) intercepts the request and prompts for a username and password, using the NetPoint Basic challenge method.

3. WebGate passes the user's credentials to the Access Server. The Access Server checks the NetPoint directory and authenticates the user. WebGate sets an ObSSOCookie in the request.
4. The Web server forwards the user request to the WAS. The NetPoint TAI gets the request and confirms that the user has been authenticated.
5. WAS recognizes that NetPoint has authenticated the user and creates an LTPA token.

---

**Note:** Remaining steps in this process are the same as steps 5. through 7. depicted in Scenario 1 (and repeated below). The WAS and both Web servers must belong to the same domain.

---

6. The NetPoint Connector for WebSphere queries the COREid Server via WebPass for a list of groups to which the user belongs.  
  
The COREid Server checks the directory and returns the information to the NetPoint Connector for WebSphere.
7. NetPoint Connector for WebSphere returns this information to the WAS.
8. The WAS checks the deployment descriptor for a user-security or group-security role mapping. If the user or group belongs to a security role that is allowed to access the resource, the WAS allows the user to access the resource.

---

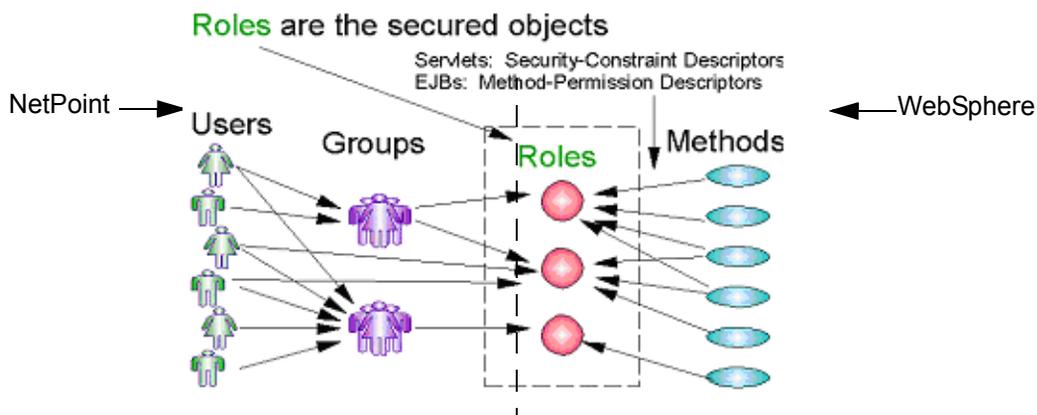
**Note:** If you are using the NetPoint CMR, step 8. doesn't occur. Instead, the Portal Server is invoked, as described in "Integration Scenario with the NetPoint CMR" on page 237.

---

## Mapping NetPoint Users and Groups to Security Roles in WAS

WebSphere security is consistent with J2EE role-based security specifications. Roles are specified in the deployment descriptors for an application. When the application is installed, these roles are bound to NetPoint users or groups. You can change the binding information for roles in an application through the WebSphere Administrative Console.

NetPoint manages users and groups. WebSphere manages roles with the help of the AAT or via the Administration Console.



In the WAS 5 Administrative Console, the role of “Admin” is special. Any NetPoint user added to the role “Admin” must be in a group called “Admin” or “Administrator”. Otherwise, this user may not be able to login to the Administrative Console. Other roles, like “monitor” have no restrictions. Therefore, any NetPoint user added to these roles in the WAS 5 Administrative Console, can log in to the WAS 5 Administrative Console.

---

**Note:** This feature is special to WAS 5 only and is not true for WAS 4.

---

# Supported Versions and Platforms

NetPoint COREid System and Access System are required for integration with WebSphere. The NetPoint Connector for WebSphere supports only the WAS Enterprise Edition. It does not support WebSphere Advanced Single-Server, the developer version. Specific support requirements are described below:

The NetPoint 7 Connectors for WebSphere include the NetPoint Custom Member Repository (CMR) for the WebSphere Portal Server. The NetPoint 7.0.2 Connector for WebSphere supports the WebSphere Application Server v5.0 and 5.1 with the NetPoint CMR (not WAS v4).

Table 10 shows the NetPoint (COREid) support for the various versions of the WebSphere Application Server (WAS) and WebSphere Portal (WSP) installed on different operating system platforms.

For the most up-to-date platform support, see the *Oracle COREid 7.x Release Notes* and the Oracle Customer Care Web site, which includes details about Oracle COREid support policy:

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

**Table 10** COREid Versions Supported for Specific WebSphere Products on Specific Operating System Platforms

Operating System Platform	WebSphere Application Server				WebSphere Portal Server			
	4.0.4 +	5.0 (Enterprise Edition)	5.1	6.0	4.2	4.1.4 with CMR	5.0.2 with CMR	5.1 with CMR
<b>Solaris 8</b>	COREid... 6.1.1.15 6.5.2 6.5.5 7.0 7.0.1 Not 7.0.2	6.5.2 6.5.5 7.0 7.0.1 7.0.2 7.0.4	7.0.2 7.0.4		6.5.2 6.5.5 7.0 7.0.1 Not 7.0.2	6.1.1.15 6.5.5 7.0 7.0.1 Not 7.0.2	6.5.5 7.0 7.0.1 7.0.2 7.0.4	
<b>Windows 2000 (Adv. Server with SP4)</b>	6.1.1.16 6.5.2 7.0 7.0.1 Not 7.0.2	6.5.2 7.0 7.0.1 7.0.2 7.0.4	7.0.2 7.0.4		6.5.2 7.0 7.0.1 Not 7.0.2	6.1.1.16 6.5.5 7.0 7.0.1 Not 7.0.2	6.5.5 7.0 7.0.1 7.0.2 7.0.4	
<b>AIX 5.1</b>	6.1.1.12 6.5.5 7.0	6.5.5 7.0 7.0.4			6.5.5 7.0	6.1.1.12 6.5.5 7.0	6.5.5 7.0 7.0.4	7.0.4
<b>AIX 5.2</b>	6.1.1.12 6.5.5 7.0	6.5.5 7.0 7.0.3 7.0.4	7.0.3 7.0.4		6.5.5 7.0	6.1.1.12 6.5.5 7.0	6.5.5 7.0 7.0.3 7.0.4	7.0.4

**Table 10** COREid Versions Supported for Specific WebSphere Products on Specific Operating System Platforms

Operating System Platform	WebSphere Application Server				WebSphere Portal Server			
	4.0.4 +	5.0 (Enterprise Edition)	5.1	6.0	4.2	4.1.4 with CMR	5.0.2 with CMR	5.1 with CMR
RedHat Linux AS 3			7.0.4					
SUSE Linux Enterprise Server 9				7.0.4				

## Preparing to Install the NetPoint Connector

Before you can install and configure the NetPoint Connector for WebSphere, you must complete the following tasks, as introduced below.

### Task overview: Preparing to install the NetPoint Connector for WebSphere

1. Install IBM and NetPoint components, as described in “Preparing Your Environment” on page 191.
2. Configure the COREid Server after installation, as described in “Configuring the COREid System for WAS Integration” on page 192.
3. Complete Access System configuration, as described in “Configuring the Access System for WAS Integration” on page 195.
4. Set up resource protection, as described in “Configuring Resource Protection in NetPoint” on page 198.
5. Define a policy domain for the Websphere Administration Console, as described in “Defining a Policy Domain for the WebSphere v6.0 Administration Console” on page 204

## Preparing Your Environment

Preparing your environment includes installing the appropriate IBM applications and NetPoint.

### To prepare your environment for integration

1. Ensure that your environment meets the requirements under “Supported Versions and Platforms” on page 189.
2. Install and configure the following IBM applications and components using the IBM documentation for these products:
  - a) IBM WebSphere Application Server
  - b) Application Assembly Tool (AAT)
  - c) WebSphere Portal Server, if appropriate for your environment.

---

**Note:** The NetPoint CMR requires the WebSphere Portal Server.

---

3. **WebSphere Portal Server v5**—See “Supported Versions and Platforms” on page 189, then follow instructions from IBM as you complete one of the following steps below to ensure you have an up-to-date version that includes Fix Pack PQ93461:

---

**Note:** Fix Pack PQ93461 is already included incorporated in Portal 5.1.

---

- Install a new WebSphere Portal 5.0.2 instance using the installation program provided for v5.0.2.
  - Upgrade an existing installation of WebSphere Portal 5.0 to v5 .0.2.
  - Install a new WebSphere 5.1 Portal instance using the intallation program provided for v5.1.
  - Upgrade an existing installation of WebSphere Portal 5.0 or 5.0.2 to 5.1
4. Install and setup NetPoint components, as described in the *NetPoint 7.0 Installation Guide*:
    - a) COREid Server
    - b) WebPass
    - c) Access Manager
    - d) Access Server
    - e) WebGate
    - f) AccessGate
  5. Configure NetPoint for WAS integration, as outlined below:

- a) **COREid System**—See “Configuring the COREid System for WAS Integration” on page 192 for details.
  - b) **Access System**—See “Configuring the Access System for WAS Integration” on page 195 for details.
6. Configure NetPoint resource protection for WAS, as described in “Configuring Resource Protection in NetPoint” on page 198.

## Configuring the COREid System for WAS Integration

The COREid Server and WebPass are the two main components of the COREid System. WebPass is a NetPoint plug-in for Web servers. When a user requests access to a Web server resource, the WebPass redirects the request to a COREid Server, which then checks the user’s identity through the directory server.

After you install the COREid Server and a WebPass, you must setup the COREid System, as explained in the *NetPoint 7.0 Installation Guide*. After setup, you configure the COREid System for integration as outlined below.

### Task overview: Configuring the COREid System for WAS integration

1. Prepare for WebPass failover, if desired, as described in “Configuring WebPass Failover” on page 192.
2. Setup the COREid Server for WAS, as described in “Configuring the COREid Server for WAS Search Methods and the NetPointWASRegistry Admin” on page 193.

## Configuring WebPass Failover

NetPoint uses failover to maximize performance and provide uninterrupted service to end users. Failover redirects requests in the event that a server fails. You must install a WebPass plug-in on each Web server. You may want to install multiple instances of WebPass. Later you may configure these for failover.

For more information, see “Configuring Multiple WebPass Instances for the NetPoint Connector” on page 216.

## Configuring the COREid Server for WAS Search Methods and the NetPointWASRegistry Admin

The COREid Server is the NetPoint component that provides user and group information to the WAS. You must configure the COREid Server to be compatible with search methods that may be used with the WAS.

By default, NetPoint requires search strings to be a minimum of 3 characters. When integrating NetPoint with WAS, you need to configure the COREid Server to allow users to use search strings that contain *less* than three characters. The method for doing this is to configure the COREid Server to accept blank search strings for the User Manager and the Group Manager.

In addition, you will configure the account for the NetPointWASRegistry Admin user. The administrator's login will be used to make IdentityXML calls to the COREid Server. The NetPointWASRegistry Admin user does *not* need to be the NetPoint Master Administrator. For example, the NetPointWASRegistry Admin user may be the Master Identity Administrator. However, if you prefer to limit the rights, the administrator you assign must possess at least the following rights:

- View access on the class attribute of the user class
- View access on the class attribute of the Group class
- The appropriate searchbase for the user and group class
- GRANT+READ right on the class attribute of the user class.
- GRANT+READ right on the class attribute of the Group class
- View access on those attributes listed in the call. For example: loginID, group member, group dynamic filter, etc.

### To configure the COREid Server after installation

1. Open the oblixappparams.xml file and set the searchstringMinimumLength to zero:

```
COREid_install_dir\identity\oblix\apps\common\bin\oblixappparams.xml
<NameValuePair ParamName="searchstringMinimumLength" value="0"/>
```

where *COREid\_install\_dir* is the directory where you installed NetPoint COREid Server.

2. Open the groupservcenterparams.xml file and set the groupMemberSearchStringMinimumLength to zero:

```
COREid_install_dir\identity\oblix\apps\groupservcenter\bin\groupservcenter
params.xml
```

```
<NameValuePair ParamName="groupMemberSearchStringMinimumLength"
value="0"/>
```

3. Restart the COREid Server.

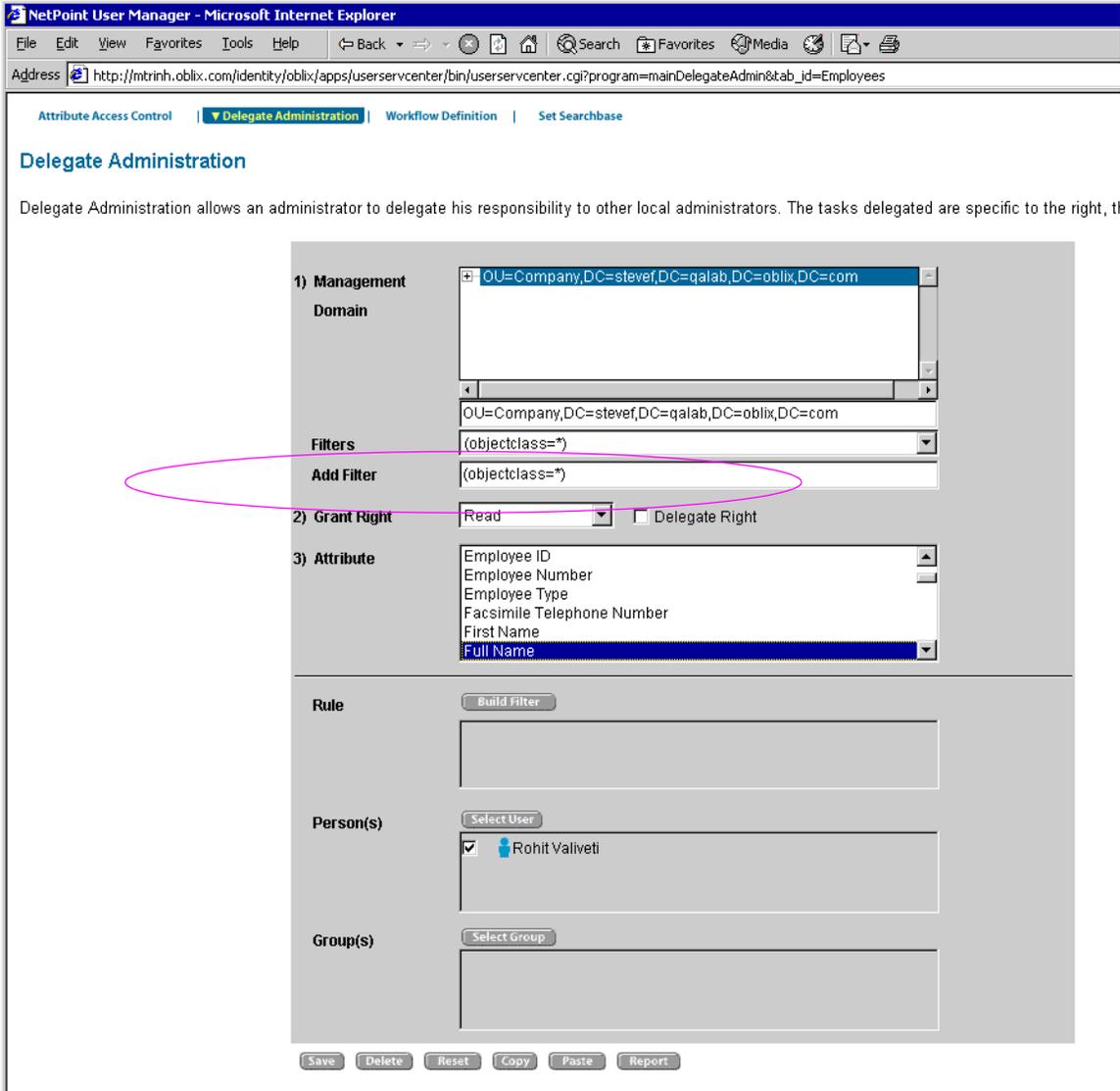
The next step must be completed after COREid System setup.

4. From the COREid System Console, you must create an administrator with the required View and Delegated Administration rights, as shown next.

---

**Important:** Do not check the Delegate Right box beside Grant Read Right.

---



5. Restart WAS and the WAS Admin Console to have the change take effect.

For more information about configuring administrators, see the *NetPoint 7.0 Administration Guide Volume 1*.

## Configuring the Access System for WAS Integration

The Access System has three main components: the Access Manager, the Access Server, and the WebGate. Installation and configuration considerations for Access System components in a WAS integration are discussed here.

**Access Manager**—The first component you install is the Access Manager. You use the Access Manager to create and manage policy domains to protect resources, and to test policy enforcement. The Access System Console is a part of the Access Manager. The Access System Console is used for system configuration and system management tasks such as configuring administrators, managing logs, and configuring instances of AccessGates and Access Servers.

**Access Server**—You also must install at least one Access Server. However, it is recommended that you install at least *two* Access Servers on two different machines to ensure uninterrupted service to your users.

Each Access Server can be configured to communicate with one or more WebGate instances, and to communicate with a directory server. Access Servers record their activity in Greenwich Mean Time (GMT) because you could have servers operating in several time zones.

**WebGate**—The WebGate component is optional. You will need it if you want to support single sign-on and if you want to protect the WebPass.

**AccessGate**—An AccessGate is a NetPoint component that processes Web and non-Web resource requests from users or applications. The NetPoint Connector for WebSphere uses an AccessGate to communicate with the Access Server.

---

**Note:** For more information, see “Configuring the AccessGate for WAS Integration” on page 195 and the *NetPoint 7.0 Administration Guide Volume 2*.

---

## Configuring the AccessGate for WAS Integration

Before installing the NetPoint Connector for WebSphere, you must install and configure an AccessGate. WebSphere intercepts user requests and passes them on to the NetPoint Connector for WebSphere. The Connector uses the AccessGate to make calls to the Access Server for authentication and authorization of the requests.

### To configure the AccessGate for the NetPointWASRegistry

1. Navigate to the AccessGate Configuration page:  
Access System Console > Access System Configuration > AccessGate Configuration
2. Click the Add button to display the Add a new AccessGate page.

3. Complete the following information:

- **AccessGate Name**—Unique, descriptive name for this AccessGate. Do not include spaces in the name.
- **Hostname**—Name of the machine where the AccessGate will be installed.
- **Port**—You do not need to specify a port. An AccessGate, unlike a WebGate does not require a port number. See the *NetPoint 7.0 Administration Guide Volume 2* for details.
- **AccessGate Password and Re-type AccessGate Password**—Unique password to verify and identify the component regardless of the transport security mode. This should differ for each AccessGate instance.
- **Access Management Service**—This service only needs to be enabled if the Access Server that you are associating with this AccessGate has Access Management Service set to On.
- **Transport Security**—Level of transport security to and from the Access Server. The default value is Open. The transport security mode of the AccessGate must match the transport security mode of the Access Server.

Your screen should look something like the one shown below. You can add other parameters now or later, as discussed in the *NetPoint 7.0 Administration Guide Volume 2*.

Oblix • NetPoint    System Configuration    NetPoint System Management    Access System Configuration

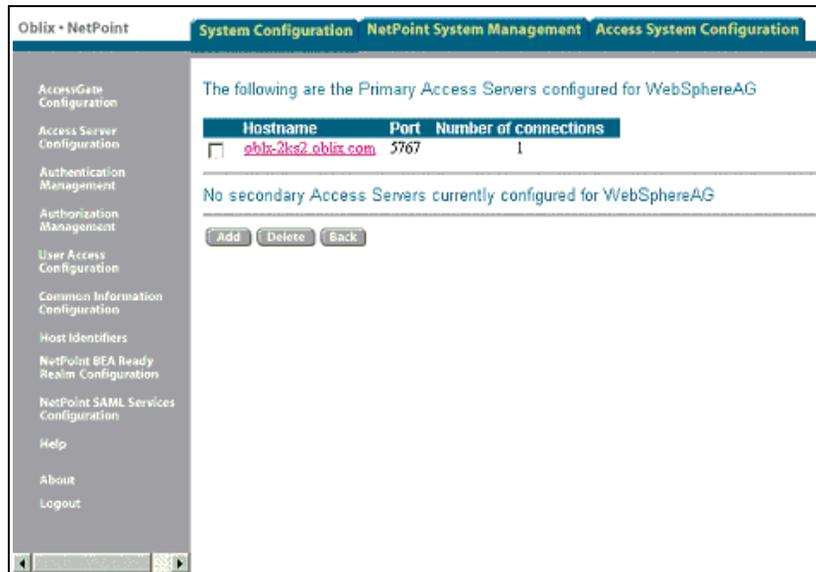
**Add a new NetPoint AccessGate**

AccessGate Configuration	<b>AccessGate Name</b>	WebSphereAG
Access Server Configuration	<b>Hostname</b>	machine.xyz.com
Authentication Management	<b>Port</b>	
Authorization Management	<b>Access Gate Password</b>	*****
User Access Configuration	<b>Re-type Access Gate Password</b>	*****
Common Information Configuration	<b>Debug</b>	<input checked="" type="radio"/> Off <input type="radio"/> On
Host Identifiers	<b>Access Management Service</b>	<input checked="" type="radio"/> Off <input type="radio"/> On
NetPoint ESA Ready Basic Configuration	<b>Maximum user session time (seconds)</b>	3600
NetPoint SAML Services Configuration	<b>Idle Session Time (seconds)</b>	3600
Help	<b>Primary HTTP Cookie Domain</b>	
About	<b>Preferred HTTP Host</b>	
Logout	<b>Maximum Connections</b>	1
	<b>Transport Security</b>	<input checked="" type="radio"/> Open <input type="radio"/> Simple <input type="radio"/> Cert
	<b>Maximum Client Session Time (hours)</b>	24
	<b>Fallover threshold</b>	
	<b>Access server timeout threshold</b>	
	<b>Sleep For (seconds)</b>	60
	<b>Maximum elements in cache</b>	100000
	<b>Cache timeout (seconds)</b>	1800

Save Cancel

The Details for the NetPoint AccessGate page appears.

4. Click the List Access Servers (or List Clusters) button to associate this AccessGate with the appropriate Access Server.
5. Click the Add button to add a new Access Server to associate with this AccessGate.
6. Select an Access Server from the drop-down list, and define the configuration for this Access Server.
7. Review the page that is returned, as shown below.



For more information about AccessGates, see the *NetPoint 7.0 Administration Guide Volume 2*.

## Configuring Resource Protection in NetPoint

The following procedures must be completed to configure NetPoint to protect resources for WebSphere.

### Task overview: Configuring resource protection in NetPoint

1. Identify a resource type, as described in “Defining a Resource Type for WebSphere” on page 199.
2. Create an authentication scheme, as described in “Defining an Authentication Scheme for WebSphere” on page 200.
3. Create a policy domain in NetPoint, as described in “Defining a Policy Domain for WebSphere” on page 201.
4. For WebSphere Application Server v6.0, create a policy domain in COREid, as described in “Defining a Policy Domain for the WebSphere v6.0 Administration Console” on page 204.

## Defining a Resource Type for WebSphere

Define a resource type for WebSphere. The procedure below guides you. See the *NetPoint 7.0 Administration Guide Volume 2* for more information on defining resource types.

In the following procedure you must provide the resource type values *exactly* as specified. If at any time you need to specify a different resource name, you must change the resource name in the following locations:

`NPCWS_install_dir\oblix\config\NetPointWASRegistry.properties`

where `NPCWS_install_dir` is the directory where the NetPoint Connector for WebSphere was installed. And:

`WAS_install_dir\properties\webgate.properties`

where `WAS_install_dir` is the directory where the WebSphere Application Server is installed.

By default, these configuration files assume you entered the values specified in the following procedure.

### To define a resource type for WebSphere

1. From the Access System Console, click Access System Configuration > Common Information Configuration
2. Click Resource Type Definitions.  
The Details for Resource Type page appears.
3. Define and save the resource type as follows:
  - **Resource Name**—Authen
  - **Display Name**—WebSphere Authentication Scheme
  - **Resource Matching**—Case Insensitive
  - **Resource Operation**—LOGIN.
4. Restart the Access Server to make this new resource available.

## Defining an Authentication Scheme for WebSphere

You need to define an authentication scheme for WebSphere. The authentication scheme provides a method to be used when determining whether a user is allowed to access a NetPoint-protected WebSphere resource.

---

**Note:** See the *NetPoint 7.0 Administration Guide Volume 2* for more information on Authentication Schemes.

---

### To define an authentication scheme for WebSphere

1. From the Access System Console, click Access System Configuration > Authentication Management.
2. Define and save the authentication scheme as shown below.
  - **Name**—WebSphere Basic Over LDAP
  - **Description**—Used to protect WebSphere-related URLs
  - **Level**—Enter a number for security level that is lower than or equal to the level specified in the authentication scheme protecting the WebSphere and Portal Server URLs. For details, see “Configuring the TAI for WebSphere 4 and NetPoint” on page 226 or “Configuring the TAI for WebSphere v5” on page 251 or “Configuring the TAI for WebSphere v6” on page 279.
  - **Challenge Method**—Basic
  - **Challenge Parameter**—Set the challenge parameter for the user credentials that you want to map.
  - **SSL required**—No
  - **Challenge Redirect**—Enter information in this field if you are going to redirect the end user's request to another server for the authentication process. The most common use of this field is to redirect from a non-SSL server to an SSL server. Redirection is transparent to the end user.
  - **Plug-ins**—Select NetPoint plug-ins to create a customized challenge scheme. For example, the NetPoint Basic Over LDAP challenge scheme requires the `validate_password` and `credential_mapping` plug-ins.

Your screen should look something like the one below.

The screenshot shows the 'Details for Authentication Scheme' page in the NetPoint Administration Guide. The page is titled 'WebSphere Basic' and provides the following configuration details:

- Name:** WebSphere Basic
- 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:** (empty)

Below these details is a table for 'Plugin(s)' with the following entries:

Order	Plugin Name	Plugin Parameters
1	credential_mapping	obMappingBase="dc=www,dc=oblix,dc=com",obMappingFilter="(&(&(objectclass=wwworgperson)(uid=%userid%))(! (obuseraccountcontrol=*))X(obuseraccountcontrol=ACTIVATED))!"
2	validate_password	obCredentialPassword="password"

At the bottom of the page, there is a checkbox for 'Update Cache' which is checked, and two buttons: 'Modify' and 'Back'.

Now you need to create a policy domain for the WebSphere Application Server, as described next.

---

**Note:** See the *NetPoint 7.0 Administration Guide Volume 2* for details on policy domains.

---

## Defining a Policy Domain for WebSphere

You need to use NetPoint to create a policy domain for WebSphere. This policy domain identifies the authentication scheme that WebSphere will use to protect the resource type that you configured in “Defining a Resource Type for WebSphere” on page 199. You also define an action in NetPoint. This action creates a user attribute for the WebSphere Application Server. When a user is authenticated to WebSphere, the user ID defined on the action is sent to the WebSphere Application Server.

### To create a policy domain for WebSphere

1. From the Access Manager, click Create Policy Domains.
2. Click the General tab and enter and save the information for your organization. For example:
  - **Name**—WebSphere

- **Description**—Policy Domain Used for WebSphere
  - **Enabled**—Yes
  - **Update Cache**
3. Click the Resources tab, then enter and save the information below for your organization. For example:
    - **Resource Type**—WebSphere Authentication Scheme  
This is the resource you defined earlier. If you changed the name of the WebSphere resource type, ensure that you specify it here.
    - **URL Prefix**—/Authen/Basic
    - **Description**—NetPointWASRegistry uses this resource for authenticating users in NetPoint. Do not delete this resource.
  4. Click the Default Rules tab > Authentication Rules > Add to create and save a default authentication rule using the WebSphere Basic Authentication Scheme that you defined earlier.
    - **Name**—WebSphere
    - **Description**—Default authentication rule using the WebSphere Basic Authentication Scheme.
    - **Authentication Scheme**—Basic Over LDAP type  
You must create a Basic Over LDAP type of authentication scheme. The Access Server SDK uses only this type of authentication scheme. You can create a new authentication scheme with a different name for WebSphere, but the authentication scheme type must be Basic Over LDAP.
  5. Click the Authorization Rules tab, then create and save an authorization rule to allow access to WebSphere resources. For example:
    - a) Click General, then enter and save:
      - **Name**—Allow Everyone.
      - **Description**—Allow access to WebSphere resources.
      - **Enabled**—Yes
      - **Allow takes Precedence**—Yes  
By default, nobody is allowed. Changing the default to Allow Everyone enables the Access Server to check a user’s identification for authentication.
    - b) Click Actions (Authorization Success), then enter and save:
      - **Return Type**—WAS\_REGISTRY

- **Name**—uid

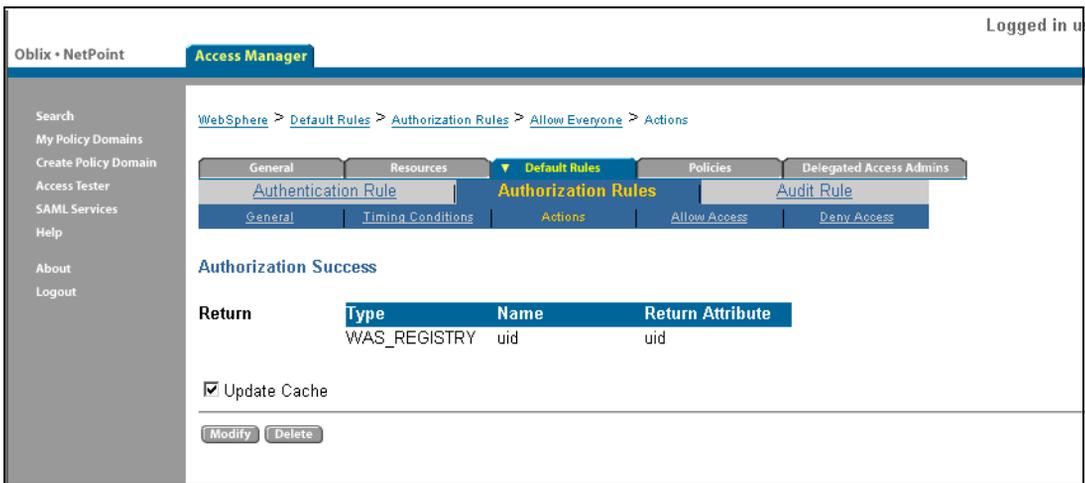
This is a hard-coded value. Use this exact string.

- **Return Attribute**—login attribute.

This attribute must be the same as the attribute configured for the login semantic type in the COREid Server or a unique attribute in the user's profile, such as uid.

c) Click Allow Access, then enter and save:

- **Role**—Anyone



6. Click OK to create the new policy domain.

You can now install the NetPoint Connector for WebSphere.

## Defining a Policy Domain for the WebSphere v6.0 Administration Console

For Websphere 6.0, you must use COREid to create a policy domain for the WebSphere administration Console. This domain protects the console URLs in the `ibm/console` and `/admin` folders, and it is required when TAI is enabled for the WebSphere AppServer. The domain helps set the TAI cookie required by the TAI component, and it also redirects the access URL to the application server Administration Console (port 9060). Under these conditions, you must also set an Authorization Success URL

### To create a policy domain for the WebSphere Administration Console

1. navigate to Access Manager > Create Policy Domains > General.
2. Enter and save the following information for your organization.

#### For example:

- **Name:** Protect WebSphere Admin Console
  - **Description:** Policy Domain Used for the WebSphere Administration Console URL
3. Click the Resources tab, then enter and save the following information for your organization

#### For example:

- **Resource Type:** http
  - **Host Identifier:** The host identifier defined for the machine hosting the WebServer.
  - **URL Prefix:** `/admin` and `ibm/console`
  - **Description:** Used by NetPointWASRegistry TAI component.
4. Click the Authorization Rules tab, then create and save an authorization rule to allow access to WebSphere Admin Console resources

#### For example:

- a) Click General, then enter and save:
  - **Name:** Allow Administrator.
  - **Description:** Allow access to WebSphere Admin Console resources.
  - **Enabled:** Yes

- **Allow takes Precedence:** Yes

By default, nobody is allowed. Changing the default to only Administrator User enables the Access Server to check a user's identification for authentication.

- b) Click Actions (Authorization Success), then enter and save the following:

- Redirect to: `http://hostname:9060/ibm/console`

If the Client Cert Authentication Scheme is to be used, set the associated variables as:

- Redirect to: `https://hostname:9043/ibm/console`

Where hostname is the fully qualified domain name of the machine hosting the WebSphere AppServer.

- c) Click Allow Access

- Select the User who has Administrative rights, then click Save.

5. To create a default authentication rule using the desired Authentication Scheme, (Basic Over LDAP, Form Based, Client Cert), navigate to Default Rules > Authentication Rules > Add . Enter and save the following values:

- **Name:** WebSphere Default Scheme
- **Description:** Default authentication rule for Admin Console.
- **Authentication Scheme:**

Create this scheme as a Basic Over LDAP / Form Based /Client Cert authentication scheme.

6. To create an authorization expression using the “Allow Administrator” authorization rule created previously, navigate to Default Rules > Authorization Expression > Add, then click Select Authorization Rule - “Allow Administrator”. Next, click Add and Save

7. Install the NetPoint Connector for WebSphere.

8. After you install and set up the connector and have also enabled TAI, enable the policy by completing the following steps:

- Navigate to General > Modify.
- Set Enable to YES, then click Save.

At this point, you can access the WebSphere Administration Console through either of the following URLs:

`http://WebServerFQDN:port/admin`

`http://WebServerFQDN:port/ibm/console`

where *WebServerFQDN* is the fully qualified domain name of the machine hosting WebServer and *port* is the port number used by WebServer.

# Installing the NetPoint Connector for WebSphere

After completing all prerequisites described above, you are ready to install the NetPoint Connector for WebSphere, as described here.

---

**Important:** If you plan to include the WebSphere Portal Server in your integration, you must install and configure the portal before you install the NetPoint Connector for WebSphere. See “Preparing Your Environment” on page 191.

---

## Task overview: Installing the NetPoint Connector

1. “Launching the Installation” on page 206
2. “Defining the Installation Directory” on page 207
3. “Specifying Connector Details” on page 207
4. “Completing Details for the WebGate” on page 209
5. “Specifying AccessGate Details” on page 212
6. “Installing a Certificate” on page 215
7. “Configuring Multiple WebPass Instances for the NetPoint Connector” on page 216

## Launching the Installation

The initial installation and setup procedure differs depending on the platform on which you are installing NetPoint.

### To launch installation

1. Insert the NetPoint CD.

The DemoShield launches.

2. Complete the step below to launch the program according to your platform:

**Windows**—Navigate to Install Oblix NetPoint > Access System > Connector for WebSphere.

**Unix**—Complete the steps below:

- Navigate to /Software/Solaris/AccessSystem/Connector for WebSphere.
- Execute /NetPoint x.x\_EN\_sparc-s2\_Connector\_for\_WebSphere.

The install wizard launches.

## Defining the Installation Directory

In this sequence, you will accept the terms of the license agreement and identify the installation directory for the NetPoint Connector for WebSphere.

You need to specify the installation directory for the NetPoint Connector for WebSphere on the machine where you installed WAS.

### To define the installation directory

1. Click Next to dismiss the Welcome Screen.
2. Read and accept the terms of the license agreement, then click Next to continue.
3. Respond to the next question based upon your platform. For example:
  - **Windows**—Click Next if you are logged in with administrator rights (otherwise click Cancel, log in as a user with administrator privileges, then restart the installation).
  - **Unix**—Specify the username and group, then click Next.

Typically, the defaults are “nobody”.

You are asked to specify the installation directory for the COREid Server. When you do this and click Next, the installation will begin and you will not be able to return to restate the name.

4. Accept the default directory by clicking Next (or change the destination, then click Next).

For example:

*\NetPoint*

You are informed that the connector is being installed, which may take several seconds.

The Configuration for NetPoint Connector for WebSphere screen appears.

## Specifying Connector Details

In this sequence you need to specify WebPass details. For failover purposes, you can configure multiple WebPass instances for the NetPoint Connector for WebSphere. You can do this either during installation or through the NetPointWASRegistry.properties file.

To configure multiple WebPass instances during installation, when you enter the WebPass hostname and port number be sure to use a comma as a separator. The hostname must be fully qualified with the domain name. For example:

**Hostname**—foo.domain.com, bar.domain.com

**Port Number**—80, 81

where the valid WebPass *host:port* combinations are:

- foo.domain.com:80
- bar.domain.com:81

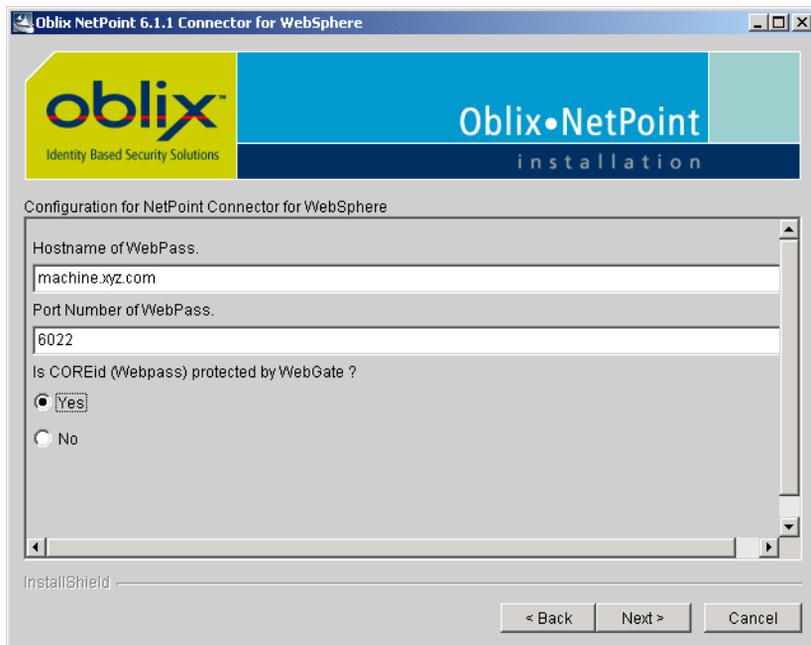
For details about configuring multiple WebPass instances through the properties file, see “NetPointWASRegistry.properties” on page 283.

## To specify NetPoint Connector for WebSphere details

1. Enter the information requested:

- **Hostname of WebPass**—The fully qualified name of the machine on which WebPass is installed.
- **Port Number of WebPass**—The port number for WebPass.
- **Is COREid (WebPass) protected by WebGate**—Specify whether the COREid (WebPass) is protected by a WebGate.

Your screen may look like the one below.



2. Click Next.

If the COREid WebPass is protected by a WebGate, the NetPoint WebGate configuration screen appears.

## Completing Details for the WebGate

You complete the WebGate configuration screen to supply the information requested.

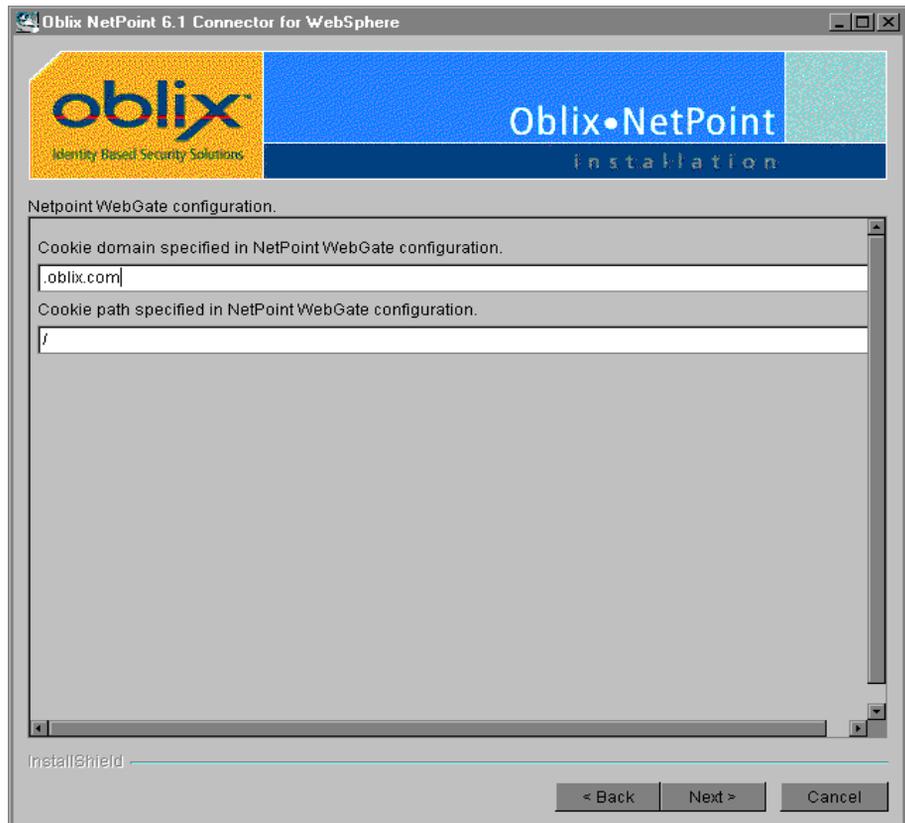
### To complete WebGate configuration details

1. Enter the cookie domain for the WebGate (for example, *.domain.com*).  
The ObSSOCookie is then recognized by all servers within this domain.
2. Enter the cookie path (/).

---

**Note:** If you have chosen to use WebGate to protect WebPass, the assumption is that you are protecting the NetPoint applications with policy domains. Therefore, it is also assumed that single sign-on between these components has been configured correctly.

---



3. Click Next.

4. Specify whether WebPass requires an HTTPS connection.

This is the SSL for secure connection when WebPass runs on HTTPS.

5. Specify the user attribute.

This attribute must be the same as the attribute configured for the Login semantic type in the COREid Server or a unique attribute in the user's profile such as uid.

6. Specify the user search attribute.

This attribute must be the same as the attribute configured for the DN Prefix semantic type for the person object class in the COREid Server. The person object class type must a structural object class. The administrator of your directory server sets this search attribute.

---

**Note:** The user attribute and the user search attribute cannot be the same.

---

7. Specify the group search attribute.

This attribute must be the same as the attribute configured for the DN Prefix semantic type for the group object class in the COREid Server. The group object class a structural object class. The administrator of your directory server sets the group search attribute.

8. Select Yes to specify that you want NetPoint Connector for WebSphere's jar files to be copied from the NetPoint Connector for WebSphere installation directory to:

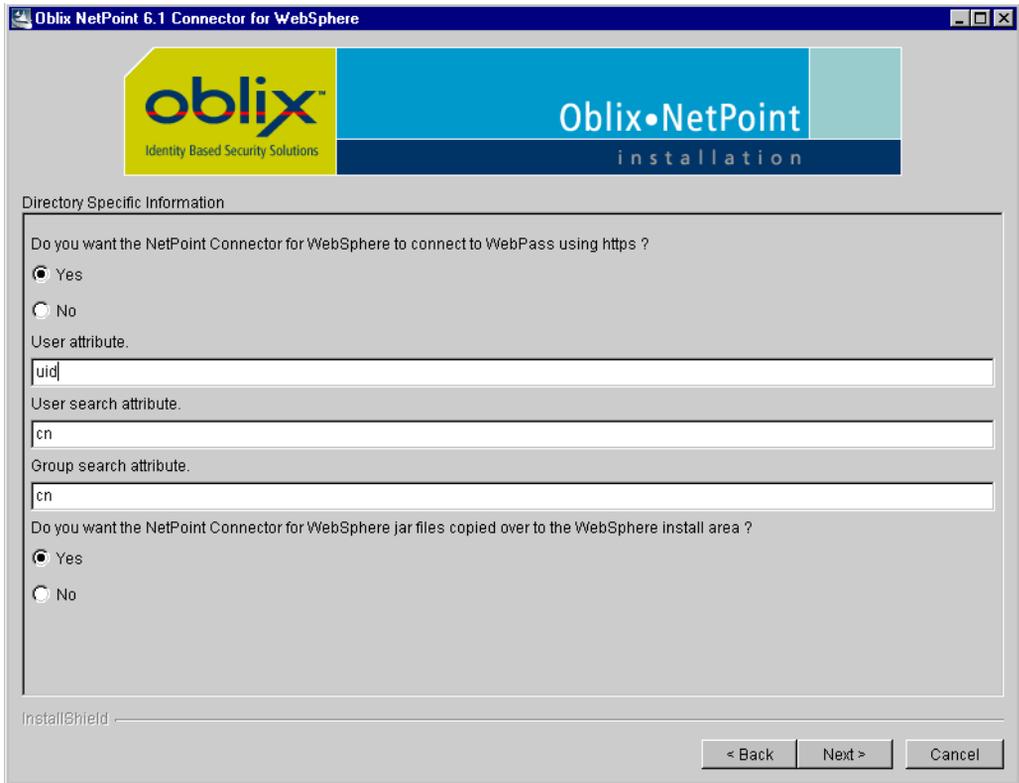
*WAS\_install\_dir/lib*

where *WAS\_install\_dir* is the directory where you installed WAS.

---

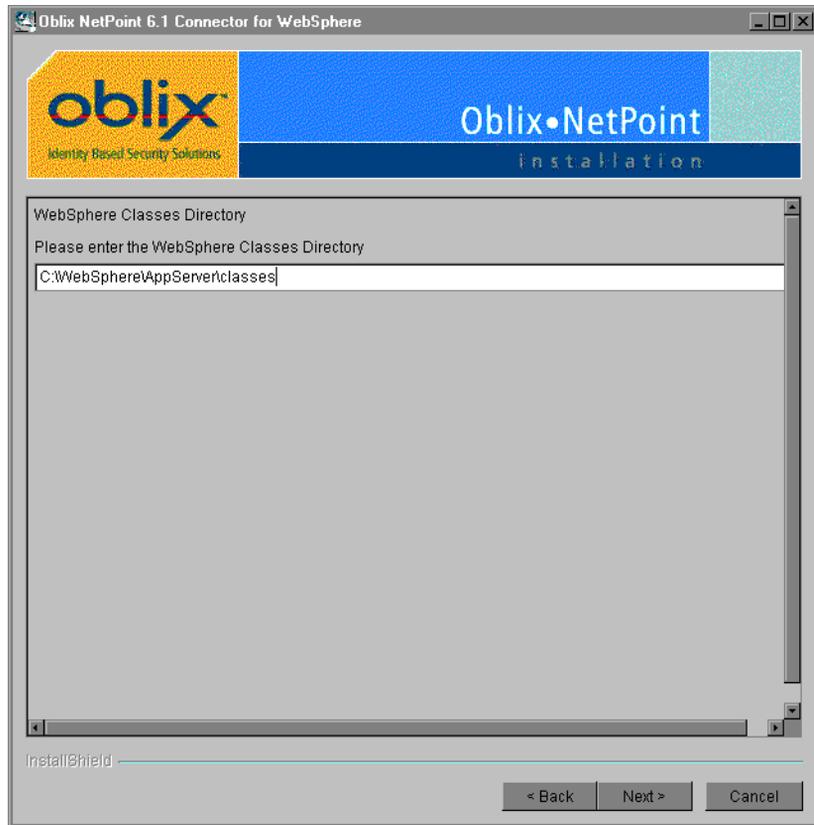
**Note:** If you select No, you must copy the jobaccess.jar and NetPointWASRegistry.jar files manually to the *WAS\_install\_dir/lib* after this installation. Or, add the location of these jar files to the WebSphere runtime classpath.

---



9. Click Next.

The WebSphere Classes Directory screen appears.



## Specifying AccessGate Details

During this sequence, you will specify the transport security mode for the AccessGate and other AccessGate details. The procedure below guides you. See the *NetPoint 7.0 Administration Guide Volume 2* for more information on AccessGates.

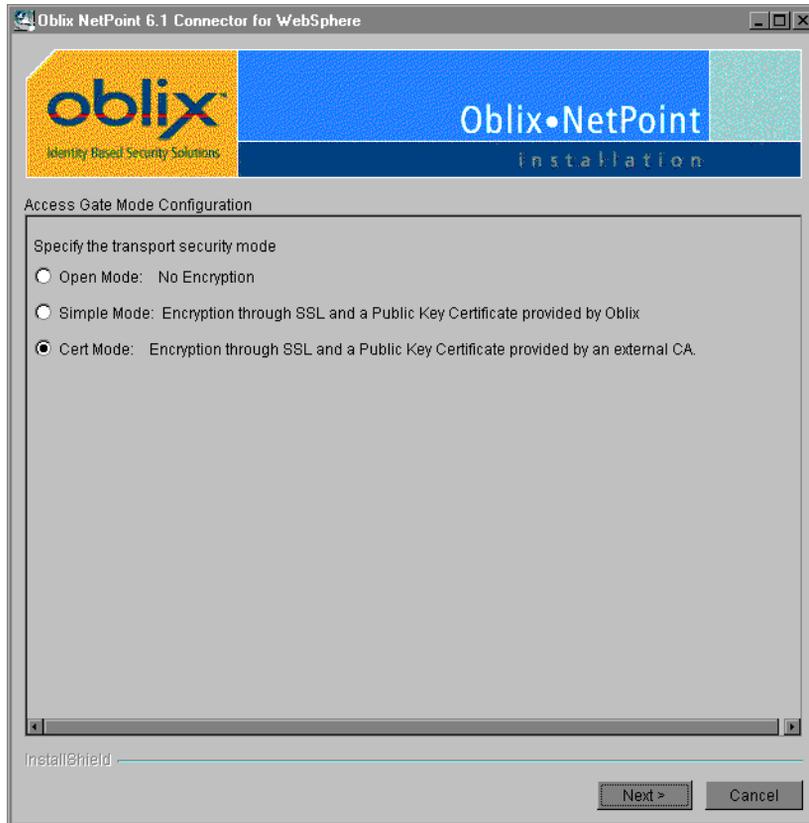
### To specify AccessGate details

1. Select the Transport Security Mode, which must be the same mode that is specified for the Access Server.

---

**Note:** Subsequent screens vary depending on the selected Transport Security Mode. See the *NetPoint 7.0 Installation Guide* for information on each selection.

---



2. Click Next.
3. Enter the following information for the AccessGate:
  - **AccessGate ID**—Enter the name you specified earlier when adding an AccessGate in the NetPoint Access System Console.
  - **Access Server ID**—Enter the name of the Access Server you associated with this AccessGate.
  - **Password for AccessGate**—Enter the AccessGate password you specified earlier when adding an AccessGate in the Access System Console, if applicable.

You can specify any Access Server associated with the AccessGate entered above.
  - **Hostname where Access Server is installed**—Enter the fully qualified hostname for the Access Server you associated with this AccessGate; for example, stontium.oblix.com.
  - **Port Number Access Server Listens To**—Enter the port of the Access Server you associated with this AccessGate.

- **Global NetPoint Access Protocol Pass Phrase**—Enter a pass phrase for all NetPoint components such as Access Server and WebGate.

This field appears only if you specify Simple Mode or Cert Mode.

- **Global NetPoint Access Protocol Pass Phrase Confirmation**—Reenter the pass phrase to confirm it.

4. Click Next.

A new AccessGate Configuration screen appears only if you specify Simple Mode or Cert Mode.

- If you need a certificate for transport security, select Request for Certificate.

NetPoint sends out a request for a certificate.

- If you already have a certificate, select Install Certificate to install it.

5. Click Next.

If you selected Install Certificate, the AccessGate Configuration screen reappears.

Oblix NetPoint 6.1 Connector for WebSphere

oblix  
Identity Based Security Solutions

Oblix•NetPoint  
installation

Access Gate Configuration

Please provide the Access Gate ID, host name, and port number for the Access Gate connection. You must use a unique ID for each Access Gate you install.

Access Gate ID  
accessgate

Access Server ID  
access-server

Password for Access Gate  
\*\*\*\*\*

Host name where an Access Server is installed  
strontium.oblix.com

Port number the Access Server listens to  
6023

Global NetPoint Access Protocol Pass Phrase  
\*\*\*\*\*

Global NetPoint Access Protocol Pass Phrase Confirmation  
\*\*\*\*\*

InstallShield

< Back   Next >   Cancel

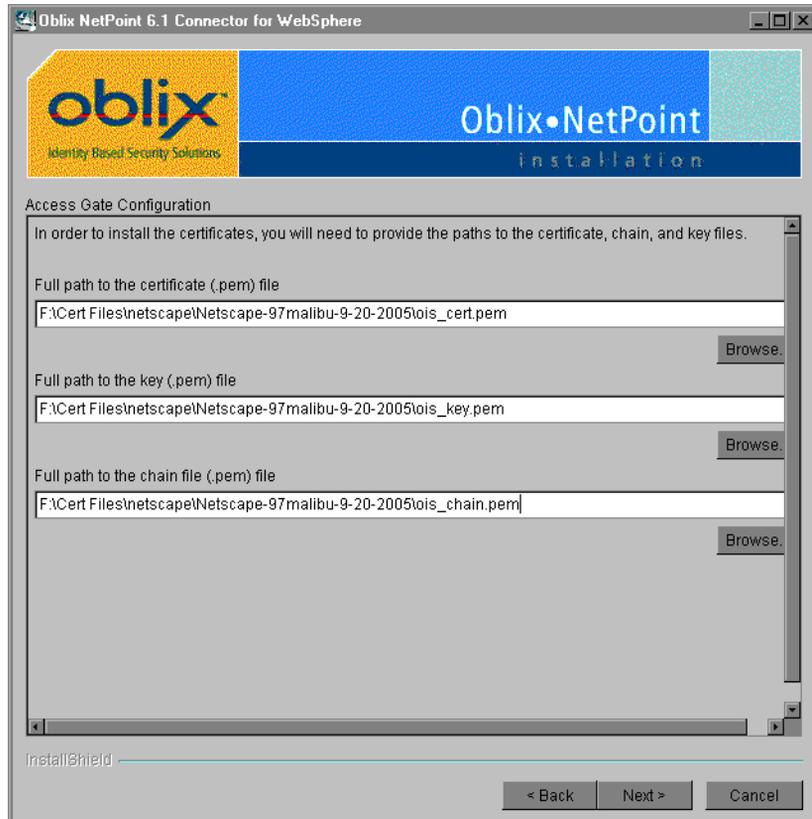
## Installing a Certificate

You must provide the paths to the certificate, chain, and key files.

### To supply the paths to the certificate files

1. Enter the full paths to the certificate, chain, and key files.

The certificate consists of these files. If necessary, click Browse to navigate to the location of these files.



2. Click Next to display the summary screen and then click Finish.

The installation is complete.

---

**Note:** If the Installation fails unexpectedly or you need to change these settings later, you can run the `configureAccessGate` tool located in `NPCWS_install_dir\oblix\tools\configureAccessGate`, where `NPCWS_install_dir` is the directory where the NetPoint Connector for WebSphere is installed.

---

Next, if necessary, configure multiple WebPass instances for NetPoint Connector for WebSphere.

## Configuring Multiple WebPass Instances for the NetPoint Connector

NetPoint uses failover to maximize performance and provide uninterrupted service to end users. Failover redirects requests when a server fails. You may want to configure multiple WebPass instances for failover purposes.

This section assumes that you have already installed more than one instance of WebPass. See the *NetPoint 7.0 Deployment Guide* for more information on failover.

### To configure multiple WebPass instances for the NetPoint Connector for WebSphere

1. Open the NetPointWASRegistry.properties file:

```
NPCWS_install_dir/oblix/config/NetPointWASRegistry.properties
```

where *NPCWS\_install\_dir* is the directory where you installed the NetPoint Connector for WebSphere.

2. Enter the fully qualified host name with the domain name and port number for the WebPass using a comma-separated list as follows:

```
# NetPoint webPass webserver host name and port number  
OB_WebPassHost=foo.domain.com,bar.doman.com  
OB_WebPassPort=81,80
```

In the above example, the valid WebPass host:port combinations are:

- foo.domain.com:81
- bar.domain.com:80

3. Complete Connector setup as described next.

# Completing NetPoint Connector Setup

After installing the NetPoint Connector for WebSphere, you must complete the following tasks to provide information about the various NetPoint components it communicates with, including the WebPass and Access Server:

## Task overview: Completing NetPoint Connector Setup

1. Complete your setup of the NetPoint Connector, as described in “Setting Up the NetPoint Connector for WebSphere” on page 217
2. Test the NetPoint Connector environment, as described in “Testing Environment Setup” on page 219

## Setting Up the NetPoint Connector for WebSphere

During the procedure below, you will ensure the jar files added during installation appear in the proper location or add the location to the WebSphere classpath.

In addition, you will ensure the environment variable path is correct. This is required because at run time, NetPointWASRegistry looks for the obaccess.dll file (Windows) or the libobaccess.so file (UNIX) that is located in the NetPoint installation directory.

### To set up the NetPoint Connector for WebSphere

1. Ensure that the following jar files that you added during installation exist in the directory *WAS\_install\_dir/lib* or add the location of these jar files to the WebSphere classpath:
  - NetPointWASRegistry.jar
  - jobaccess.jar
2. Add *NPCWS\_install\_dir\oblix\lib* to the environment variable path, as follows:

**Solaris**—In the *setupCmdLine.sh* file, add as follows:

```
NPCWS_install_dir\oblix\lib to $LD_LIBRARY_PATH
```

where *NPCWS\_install\_dir* is the directory where you installed the NetPoint Connector for WebSphere.

---

**Note:** The NetPointWASRegistry may fail to install if it can not find the Access Server SDK. You may need to add the following environment variables to the *setupCmdLine.sh*:

```
OBACCESS_INSTALL_DIR=NPCWS_install_dir.
```

---

**AIX**—In `setupCmdLine.sh` file, add as follows:

```
NPCWS_install_dir\oblix\lib to $LIBPATH
```

Restart the WebSphere Administration Server.

**Windows 2000**—The installer automatically adds the information. However, you can:

- a) Manually add `NPCWS_install_dir\oblix\lib` to the PATH System variable.
  - b) Reboot the machine.
  - c) Start the WebSphere Administration Server.
3. Check the configuration to ensure that WebGate is protecting the WebPass.
- a) Open the `WebGateStatic.lst` file located in the following directory:  
`WebGate_install_dir\access\oblix\apps\webgate`  
where `WebGate_install_dir` is the directory where WebGate is installed.
  - b) Set `IPValidation = false`

---

**Note:** If you want to set `IPValidation = true`, check the `IPValidationExceptions` list for the IP address.

---

4. Restart the Web server.
5. Verify the information in the `NetPointWASRegistry.properties` file located `NPCWS_install_dir\oblix\config`.  
where `NPCWS_install_dir` is the directory where your NetPoint Connector for WebSphere is installed. The file is populated with information that was specified during installation. For more information, see “NetPointWASRegistry.properties” on page 283.
6. Determine if the machine hosting WebPass is running SSL, and if so, complete the following steps:
  - a) Open the `NetPointWASRegistry.properties` file and set `OB_WebPassSSLEnabled = True`.
  - b) Obtain the WebServer and CA certificates of the Web server hosting WebPass or WebGate running in SSL mode and place them respectively in the `server.cer` file and the `ca.cer` file.
  - c) Use `keytool` in `JAVA_HOME\bin` or `JAVA_HOME\jre\bin` to add the following ca and server certificates to `jssecacert` keystore:
    - `keytool -import -alias ca -file ca.cer -keystore jssecacert`
    - `keytool -import -alias server -file server.cer -keystore jssecacert`

- d) Depending on the Java version that you are using, copy this file to the security directory located in *JAVA\_HOME*\lib\security, or in *JAVA\_HOME*\jre\lib\security.

The NetPoint Connector for WebSphere uses WebPass to make IdentityXML calls. You can specify only one WebPass at a time for NetPoint Connector for WebSphere. Typically, this will be on the same Web server that hosts the Access Manager. If you have more than one WebPass, and want NetPoint Connector for WebSphere configured for a different WebPass, you can change the host machine and port in NetPointWASRegistry.properties file after installation.

If you edit the NetPointWASRegistry.properties file, you must restart the WebSphere Administration Server.

---

**Note:** If WebPass is protected by a WebGate, ensure that the security level in this authentication scheme is the same level or a lower level than the one specified in the WebSphere authentication scheme discussed in “Defining an Authentication Scheme for WebSphere” on page 200.

---

## Testing Environment Setup

Before you enable the NetPointWASRegistry, you need to run the registryTester program to ensure that the NetPointWASRegistry is registered and can successfully connect to the NetPoint COREid System.

The following procedure applies to all WAS versions.

### To run the registryTester program

1. Edit the file registerTester.bat (Windows) or registryTester.sh (UNIX) in the *NPCWS\_install\_dir*/unsupported directory.

where *NPCWS\_install\_dir* is the directory where you installed the NetPoint Connector for WebSphere.

2. Modify these two variables as follows:
  - **INSTALL\_DIR**—Specify the path to the NetPoint connector for WebSphere.
  - **WAS\_INSTALL\_DIR**—Specify the path to the WAS.

In the next step, you need to comment out any classpaths that are not relevant to your installation.

3. Specify the correct classpath and comment out the unused classpath for your installation :
  - **WebSphere 4**—Keep WebSphere classpath for v4; comment out irrelevant WebSphere classpaths.

- **WebSphere 5.0 with NetPoint Connector 7.0**—Keep WebSphere classpath for v5.0; comment out irrelevant WebSphere classpaths.
  - **WebSphere 5.0.2**—Keep WebSphere classpath for v5.0.2; comment out irrelevant WebSphere classpaths.
  - **WebSphere 5.1 with NetPoint Connector 7.0.2**—Keep the WebSphere classpath for v5.1; comment out irrelevant WebSphere classpaths.
  - **WebSphere 6.0 with NetPoint Connector 7.0.4**—Keep the WebSphere classpath for v5.1; comment out irrelevant WebSphere classpaths.
4. If you do *not* have a JAVA\_HOME environment variable defined, set the JAVA\_HOME parameter value as follows in the registryTester.bat or registryTester.sh file:

**Windows**—%JAVA\_HOME% = *WAS\_install\_dir*\java

**UNIX**—\$JAVA\_HOME\$ = *WAS\_install\_dir*/java

where *WAS\_install\_dir* is the directory where you installed WebSphere.

5. **WebSphere 5.x only**—Update the registryTester.sh (.bat on Windows) as follows:

```
set CLASSPATH=.:${CLASSPATH}:${INSTALL_DIR}/oblix/lib/
NetPointWASRegistry.jar
:${INSTALL_DIR}/oblix/lib/jobaccess.jar
:${WAS_INSTALL_DIR}/lib/xerces.jar
:${WAS_INSTALL_DIR}/lib/j2ee.jar
:${WAS_INSTALL_DIR}/lib/wssec.jar
:${WAS_INSTALL_DIR}/java/jre/lib/ext/ibmjsse.jar
```

---

**Note:** You may check the registryTester.bat for details.

---

6. From the command line, run registryTester:
- **NT/W2K**—registryTester.bat
  - **UNIX**—registryTester.sh
7. Supply a NetPoint userid and password when prompted.
8. Verify the result:
- If the program completes successfully, it connects to the COREid Server and returns a list of groups to which the user belongs.
  - If the registryTester program fails to connect with the COREid Server, check the parameter values in the NetPointWASRegistry.properties file and correct them as needed.

# Configuring WebSphere Application Server v4

This discussion describes how to configure the WebSphere Application Server v4 for integration of NetPoint. If you are using the NetPoint 7.0.2 Connector, see “Configuring WebSphere Application Server v5” on page 247.

Before you start, check the discussion “Supported Versions and Platforms” on page 189 for more information:

- The NetPoint 7.0 Connector for WebSphere does not support the WebSphere Application Server v5.
- The NetPoint 7.0.2 Connector for WebSphere does not support the WebSphere Application Server v4.

## **Task overview: Configuring WebSphere 4 for the NetPoint Connector**

1. Enable the NetPointWASRegistry, as described in “Enabling the NetPointWASRegistry in WAS 4” on page 221.
2. Test authentication and role mapping, as described in “Testing NetPointRegistry for WAS v4, v5, and v6” on page 224.
3. Set up the TAI, as described in “Configuring the TAI for WebSphere 4 and NetPoint” on page 226.

## **Enabling the NetPointWASRegistry in WAS 4**

Once you have installed the products and tested them to be sure they are communicating, you can enable the NetPointWASRegistry. Enabling the registry allows NetPoint to be used as the authentication source for the WebSphere Application Server. This enables the NetPointWASRegistry, which works with the CustomRegistry in WebSphere 4. See:

- “Specifying NetPointWASRegistry in WAS 4” on page 222
- “Specifying an Administrative Role” on page 224

## Specifying NetPointWASRegistry in WAS 4

To specify NetPointWASRegistry in WebSphere, you must change the mode from LDAP to Custom User Registry in the Security Center window and specify an administrative role for the NetPointWASRegistry. An administrative role authorizes a user to perform administration tasks.

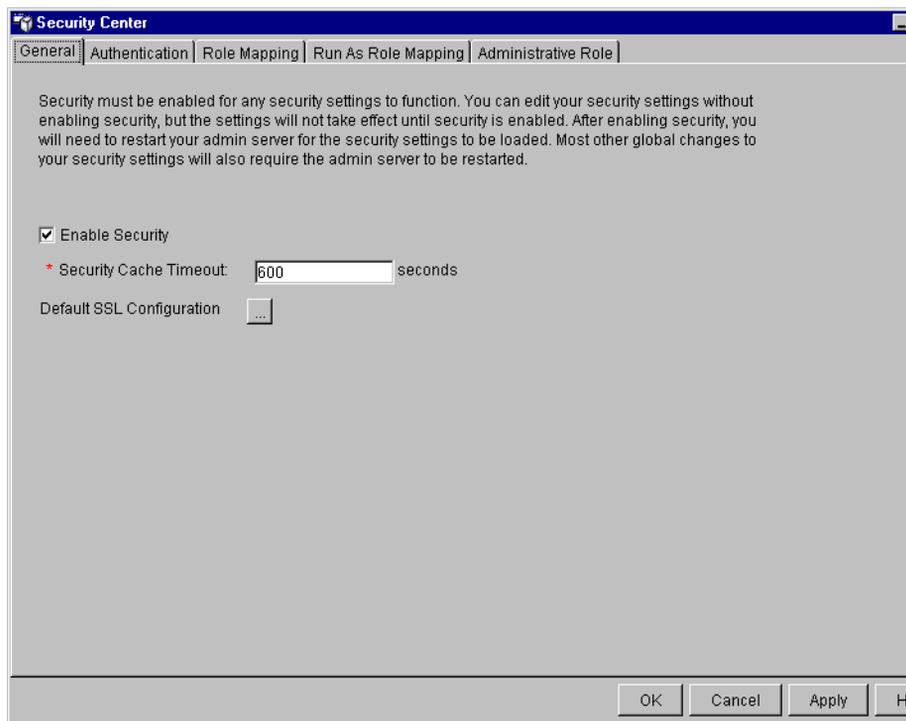
---

**Note:** It is assumed that the current mode is LDAP and that security is enabled for this mode.

---

### To enable the NetPointWASRegistry

1. Launch the WebSphere Administrative Console.
2. Open Console > Security Center.
3. In the General Tab, ensure that Enable Security is already checked.



4. In the Authentication Tab, select Lightweight Third Party Authentication (LTPA).
5. Enable single sign-on
6. In the Domain field, enter the correct domain name; for example, .oblix.com.

---

**Note:** Be sure to include the leading “dot,” for example, .oblix.

---

7. Select Custom User Registry.

The Custom User Registry window opens.

8. In the Security Server ID field, enter the name of the Delegated administrator configured in NetPoint.

This user should have the right to view users and groups. NetPoint Connector for WebSphere connects to the COREid server as this user to retrieve user and group data.

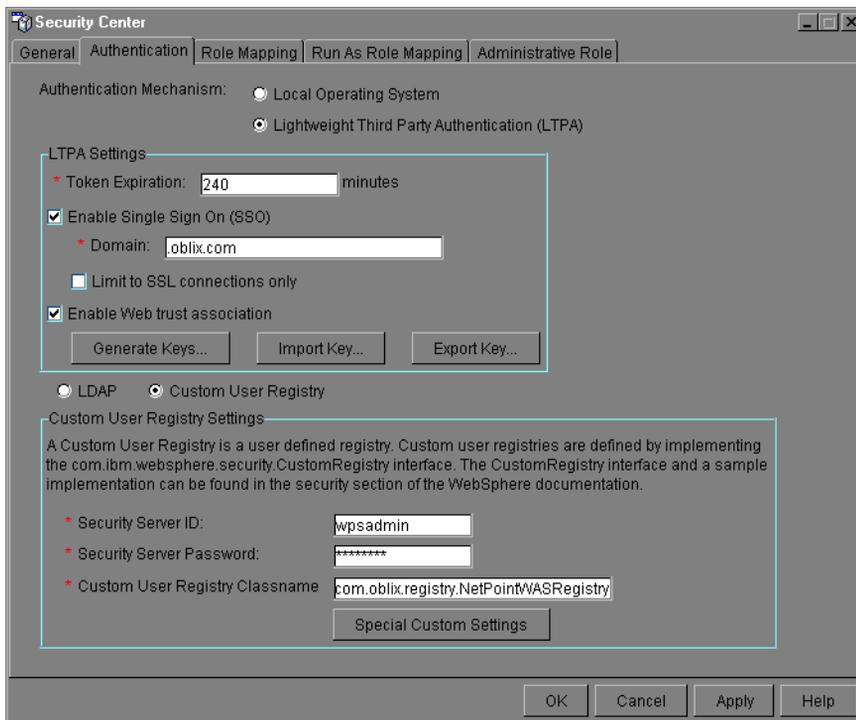
9. In the Security Server Password field, enter the Delegated administrator's password.

---

**Note:** Whenever the Security Server Password is changed in Netpoint, a corresponding change should also be made in the custom registry settings from the Websphere Administrative Console Security Center.

---

10. In the Custom User Registry Classname field, enter com.oblix.registry.NetPointWASRegistry.



11. Click Special Custom Settings.

The Specify Custom Settings screen appears.

12. In the Name field, enter NetPointWASRegistry.properties.

13. In the Values field, specify the file name and the full path to the NetPointWASRegistry.properties file.

For example:

```
c:\NPCWS_install_dir\oblix\config\NetPointWASRegistry.properties
```

---

**Note:** You must specify the full path.

---

14. Click Apply and then click OK.

A message appears stating that the security changes will take place after the Administration Server is started.

You must now specify an administrative role for the NetPointWASRegistry.

## Specifying an Administrative Role

Next, you need to specify an administrative role for the NetPointWASRegistry.

### To specify an administrative role for NetPointWASRegistry

1. Restart the Administration Server.  
The NetPointWASRegistry challenges you for your credentials.
2. Enter your credentials.
3. Open Console > Security Center.
4. In the Administrative Role tab, deselect all the specified users and groups because they belong to the LDAP mode.
5. Search for the users and groups that you want to specify in the administrative role for NetPointWASRegistry.

---

**Note:** A search string entered in the WebSphere Administrative Console should end with an asterisk (\*); for example, admin\*. If \* is not included, the search will not return any results.

---

6. Save your changes.
7. Restart the Administration Server.
8. See “Testing NetPointRegistry for WAS v4, v5, and v6” on page 224.

## Testing NetPointRegistry for WAS v4, v5, and v6

After you have enabled the NetPointRegistry, test it for successful authentication and role-mapping between WebSphere and NetPoint.

To conduct these tests, use the Snoop servlet that WebSphere provides. The Snoop servlet has security constraints that only allow access to authenticated users.

Ensure that the NetPointWASRegistry is authenticating NetPoint users successfully as described in “Scenario 1: Use of NetPointWASRegistry” on page 184.

Then, ensure that WebSphere security roles are being mapped correctly to NetPoint-managed users and groups.

### **To test authentication by the NetPointWASRegistry for WAS v4**

1. Access the Snoop servlet at the following URL:

`http://hostname.domain.com:9080/servlet/snoop`

where *hostname* is the fully qualified name of the machine where WebSphere is installed; for example, *xyz.domain.com*.

2. Log in as a NetPoint user.

If authentication is successful, you will be allowed to access the page. If authentication is unsuccessful, you will get a message stating that you are not authorized to view the page.

### **To Test authentication by the NetPointWAS Registry for WAS v5 and v6**

1. Launch the WebSphere Administrative Console and navigate to Applications > Enterprise Applications > DefaultApplication in the left panel.

2. Select the “Map security roles to users/groups” tab.

The two roles available in this tab are All Role and Everyone Role.

3. Select specific users and groups for each of these two roles.

These are your NetPoint-managed users and groups.

4. Restart the DefaultApplication.

5. Access the Snoop servlet at the following URL:

`http://hostname.domain.com:9080/snoop`

where *hostname* is the fully qualified name of the machine hosting WebSphere.

6. Sign in as one of the users you assigned to either of the two roles in step 3.

If role-mapping succeeds, you can access the servlet.

7. Sign in as a user who is not assigned to one of the roles assigned in step 3.

You should receive a return message announcing that you are not authorized to view the page.

## To test mapping of WebSphere roles to NetPoint users and groups

1. Launch the WebSphere Administrative Console and click Enterprise Applications in the left panel.
2. Select *hostname\_sampleApp*, where *hostname* is name of the machine where WebSphere is installed.  
For example, *phillip\_sampleApp*.
3. In *sampleApp*, click the User Role Mappings tab.  
The two roles available in this tab are All Role and Everyone Role.
4. Select specific users and groups for each of these two roles.  
These are your NetPoint-managed users and groups.
5. Restart the Default Server.
6. Access the Snoop servlet at the following URL:  
`http://hostname.domain.com:9080/servlet/snoop`  
where *hostname* is the fully qualified name of the machine where WebSphere is installed; for example, *xyz.domain.com*.
7. Sign in as a user that you assigned to either of the two roles in step 4.  
If role-mapping was successful, you will be able to access the servlet.
8. Sign in as a user who is not assigned to these roles.  
You get a message stating that you are not authorized to view the page.

## Configuring the TAI for WebSphere 4 and NetPoint

You configure the TAI to enable single sign-on between NetPoint and WAS, as well as between NetPoint and the WebSphere Portal Server.

For WebSphere 4.0, this involves copying two configuration files: `webgate.properties` and `trustedservers.properties` and updating the WAS Administration Console to use the TAI.

## To install and configure TAI for WAS 4

1. Copy the configuration file named `webgate.properties` (see Table 12 for parameters) as follows:

**From:** `NPCWS_install_dir/oblix/config`

**To:** `WAS_install_dir/properties` folder in the WebSphere installation properties directory

where `NPCWS_install_dir` is the directory where NetPoint Connector for WebSphere is installed and `WAS_install_dir` is the directory where the WebSphere Application Server is installed.

This file contains configuration information that WebSphere will use to connect to the AccessGate.

2. In the WebSphere installation properties directory, modify the parameter values of the `webgate.properties` file (Table 12) as follows:

`OB_InstallDir = NPCWS_install_dir`

For example, `C:\NetPoint\NetPointWASRegistry`

If WebGate is installed on a proxy server that is used as a front end server to direct all user requests to Web servers that interface with WebSphere Application Servers, then specify the following parameter values:

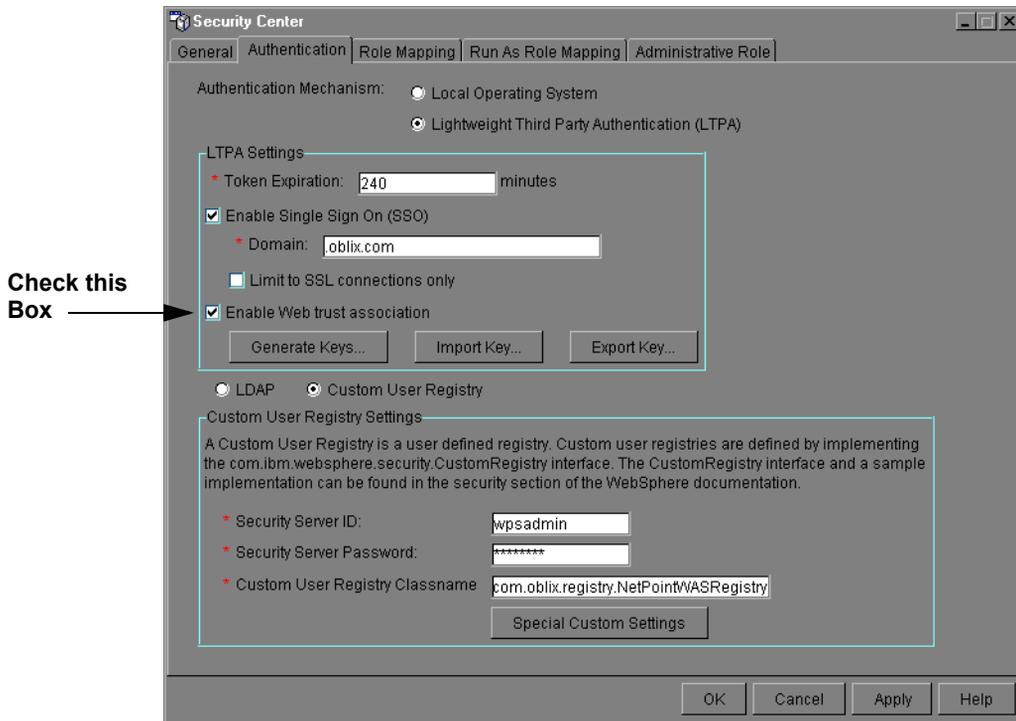
- `OB_hostnames = serverName`

where `serverName` is the name of the proxy server.

- `OB_ports = portNumber`

where `portNumber` is the port number of the proxy server.

3. Copy the contents of the `trustedservers.properties` file (Table 13) that is located in your `NPCWS_install_dir`, and append them to the `trustedserver.properties` file located in the `WAS_install_dir\properties` directory.
4. Comment out other entries such as references to WebSEAL.
5. In the Authentication tab of the WebSphere Application Server Security Console, select Enable Web Trust Association.



**6. Install a WebGate plug-in to the Web server servicing the WAS.**

WebGate is needed for single sign-on. See the *NetPoint 7.0 Installation Guide* for more information on installing WebGate.

If you have already installed a WebGate to protect the WebPass, you can use the already-installed WebGate.

**7. In the NetPoint Access Manager, define a policy for the resource that you want to protect.**

Other authorization rules can also be added at this point. The policy that you use to protect the URL can use basic, form, cert, or other authentication schemes that NetPoint supports.

Ensure that the security level in this authentication scheme is *equal* or *greater* than the one specified in the WebSphere authentication scheme discussed in “Preparing Your Environment” on page 191.

See the *NetPoint 7.0 Administration Guide Volume 2* for more information on defining policies.

- Restart the WAS.

---

**Note:** It is recommended that you use a form-based authentication scheme to protect resources. If you use the basic authentication scheme, set the Challenge Redirect field to another WebGate to ensure that the ObSSOCookie gets sent.

---

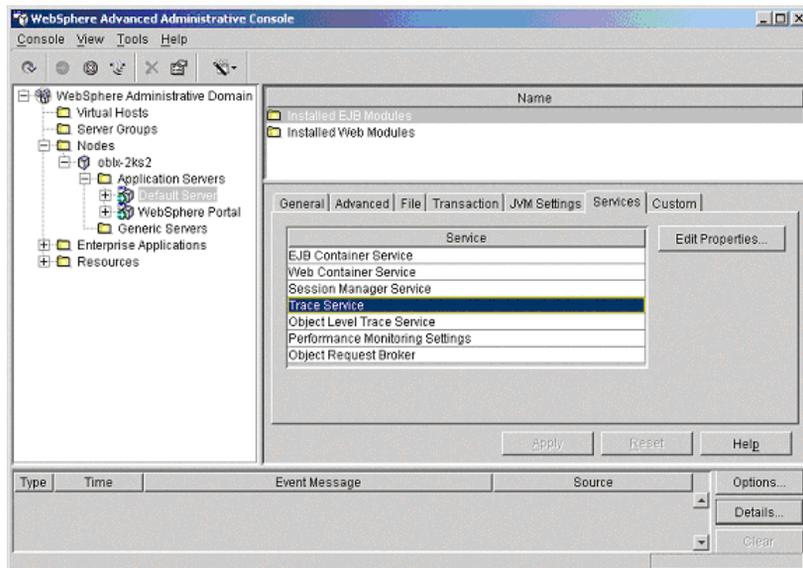
- See optional implementation details in “Implementation Notes for the TAI” on page 292.

## Enabling Logging for TAI for WAS 4

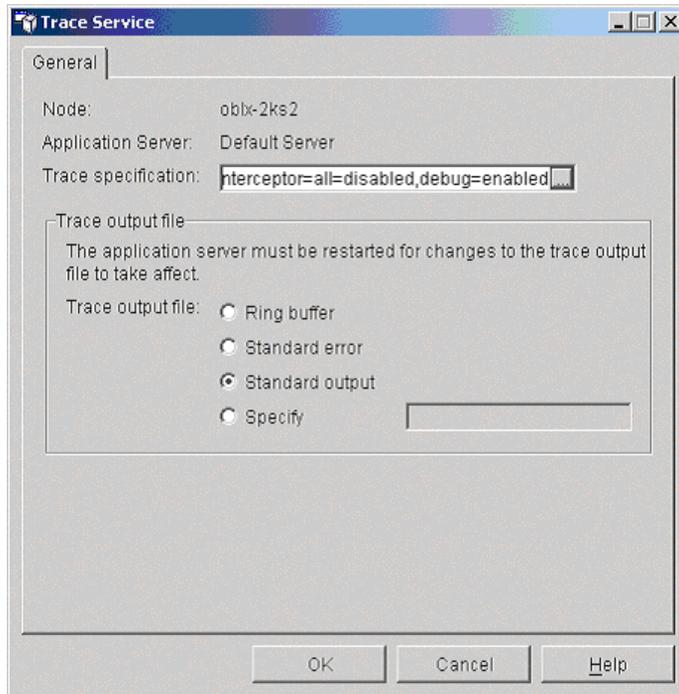
You can enable logging for the TAI from the WebSphere Administrative Console. The procedure to enable logging varies depending on the WAS version that you use.

### To enable logging for TAI for WAS 4

- Launch the WebSphere Administrative Console.
- Navigate to Nodes > *MachineName* > Application Servers > Default Server. *MachineName* is the name of the machine where you have installed WAS.



- In the Services Tab, select Trace Service, and click Edit Properties. The Trace Service window appears.



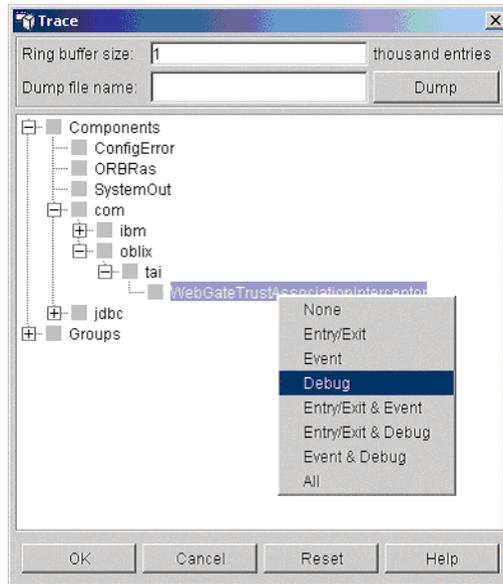
4. Select Standard output or specify a file name.

If you select Standard output, the NetPoint Connector for WebSphere TAI logging will be recorded in the *WAS\_install\_dir\logs\Default\_Server\_stdout.log* file.

5. Select a Trace Specification and click OK.

The Trace window appears.

6. Expand Components > com > oblix > tai > WebGateTrustAssociationInterceptor.
7. Right click WebGateTrustAssociationInterceptor and select the level of logging; for example, Debug.



8. If you wish, you can specify a Trace output filename.

## Testing NetPoint Single Sign-on for WAS v4.x

After you have configured TAI, test for successful authentication and single sign-on between WebSphere and NetPoint.

To conduct these tests, you use the Snoop servlet that WebSphere provides. The Snoop servlet has security constraints that only allow access to authenticated users. When WebSphere security is not enabled, access to the Snoop is unrestricted. When WebSphere security *and* TAI are enabled, users attempting to access Snoop will be challenged by NetPoint. If TAI is *not* enabled, users attempting to access Snoop will be challenged by WebSphere.

To test NetPoint single sign-on for WAS, you must build and configure a new WebSphere secure application. Then, test NetPoint authentication and single sign-on for the secure application.

During installation, a secure application is built that you can use for testing and stored in the following location:

*NPCWS\_install\_dir/examples/securityapp/SimpleSessionSecure.ear*

where *NPCWS* is the directory where you installed the NetPoint Connector.

---

**Note:** If you wish, you can build your own secure application. The following procedure describes how to build an application for WAS 4.x and 5.x.

---

## To build a WebSphere secure application

1. Build a secure application named SimpleSessionSecure according to the instructions available at:  
[http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/060704\\_security.html](http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/060704_security.html)
2. Save the SimpleSessionSecure.ear file in the appropriate location; for example, in c:\temp.
3. Verify that the WAS Administration Server is running.

## To install the SimpleSessionSecure application

1. Launch the WebSphere Administrative Console located in *WAS\_install\_dir\bin\adminclient*.  
where *WAS\_install\_dir* is the directory where the WebSphere Application Server is installed.
2. In the WebSphere Console tree view, right-click WebSphere Administrative Domain > Enterprise Applications.
3. From the resulting menu, click Install Enterprise Application to launch the Install Enterprise Application wizard.  
The Specifying the Application or Module panel appears:
4. Verify the following settings:
  - The Browse for file on node field is set to your current node.
  - The Install Application option is selected.
5. Click Browse to locate and select the SimpleSessionSecure.ear file.
6. Verify that its name is now displayed in the Path field and specify SimpleSessionSecure as the Application name.
7. Click Next, then click Yes when prompted whether to deny access to unprotected methods.
8. In the Mapping Users to Roles panel, verify that the Goodguys role is mapped to valid NetPoint Users.
9. Click Select; verify that NetPoint Users are listed in the Selected Users/Groups area of the resulting Select Users/Groups dialog, then click OK to close the dialog after verification.
10. Click Next.
11. On the Mapping EJB RunAs Roles to Users panel, click Next.
12. In the Binding Enterprise Beans to JNDI Names panel, verify that the JNDI Name is set to gs/hello, and then click Next.

13. In the Mapping EJB References to Enterprise Beans panel, verify that the JNDI Name is set to `gs/hello`, and then click Next.
14. Click Next in the next three panels, until the Selecting Virtual Hosts for Web Modules panel appears.
15. In the Selecting Virtual Hosts for Web Modules panel, ensure that the Virtual Host is set to `default_host`, then click Next.
16. In the Selecting Application Server panel, ensure that the EJB11 and SimpleSessionWar modules reside on Application Server named Default Server and then click Next.
17. In the Completing the Application Installation Wizard panel, click Finish.
18. When prompted to regenerate code, click No.
19. Look for the message confirming successful installation of the application.  
It may be a minute before it is displayed. You can now view the SimpleSessionSecure application in the WebSphere Administrative Console tree view.
20. After you build the SimpleSessionSecure application, regenerate the plug-in configuration to enable the Web server to locate the WebSphere application.

### To regenerate the plug-in configuration

1. In the console tree view, right-click WebSphere Administrative Domain > Nodes > *hostname*.  
where *hostname* is the name of the machine where WebSphere is installed.
2. From the resulting menu, select Regen Webserver Plugin.
3. In the Event Message panel, a message appears stating that the plug-in regeneration has been completed.
4. Stop the WebSphere Administrative Server and start it again:  
To stop the administrative server, under Nodes in the WebSphere Administrative Console, right-click *hostname* and select Restart from the resulting menu. The console will close.
5. Open the WebSphere Administrative Console again after the administrative server starts.  
This time, you will be asked to log in, because security is enabled.
6. In the WebSphere Administrative Console tree view, click WebSphere Administrative Domain > Nodes > *hostname* > Application Servers > Default Server.  
where, *hostname* is the fully qualified name of the machine where WebSphere is installed; for example, *xyz.domain.com*.

7. Ensure that the Module Visibility setting of the Default Server is set to Compatibility.
8. If you want to change the visibility setting, click Apply.

### **To test NetPoint authentication and single sign-on**

1. Access the SimpleSessionSecure application at the following URL:

`http://hostname/gettingstarted3/SimpleSession?msg=Hi`

where *hostname* is the fully qualified name of the machine where WebSphere is installed; for example, *xyz.domain.com*.

2. Log in as a NetPoint user.
  - **TAI Not Enabled**—If you have not enabled the TAI, you will be challenged by WebSphere and your credentials are passed on to NetPoint. After NetPoint authenticates you, you will be allowed to access SimpleSessionSecure.
  - **TAI Enabled**—If you have enabled TAI, you will be challenged and authenticated by NetPoint. Because single sign-on between NetPoint and WAS is enabled, you are allowed to access SimpleSessionSecure and other NetPoint protected resources (URLs) without being challenged by WebSphere.

### **To test NetPoint single sign-on for NetPoint-protected WebSphere resources**

1. On the Web server you use to access the WAS, navigate to the document root and create a directory named test.
2. In the test directory, create a file named index.html.
3. In NetPoint, create and enable policies to protect /servlet and /test directories.
4. Access the Snoop servlet at the following URL:

`http://hostname.domain.com:9080/servlet/snoop`

where *hostname* is the fully qualified name of the machine where WebSphere is installed; for example, *xyz.domain.com*.

You will be challenged for authentication. After you are authenticated, you will be allowed to access the Snoop servlet.

5. Access the /test URL.

You must be allowed to access the URL and view the index.html file without being challenged.

# Integrating NetPoint with the Portal v4

A portal provides a single point of access to enterprise data and applications, presenting a unified and personalized view of that information to employees, customers, and business partners.

The WebSphere Portal Server runs on top of the WAS and uses the WAS security infrastructure to enforce access control. The WebSphere Portal Server 4 currently operates only with the WebSphere Application Server 4.

Integrating NetPoint with the WebSphere Portal provides the following NetPoint functionality for the portal:

- User and group management
- Password management
- SSO to the portal
- Unified logout between NetPoint, WAS, and the WebSphere Portal
- Portlet Access Control

## WebSphere Portal Component

The WebSphere Portal Server uses the following component to manage user and group information: Member Services component.

**Member Services**—The Member Services component manages attributes for portal users. The WebSphere Portal uses Member Services to keep track of the attributes and attribute values of users and groups that are registered to the portal. Member Services help the portal to manage information on user accounts, user profile attributes, and group memberships, and portlet access control.

See the WebSphere Portal Server documentation for more information on the portal and related components.

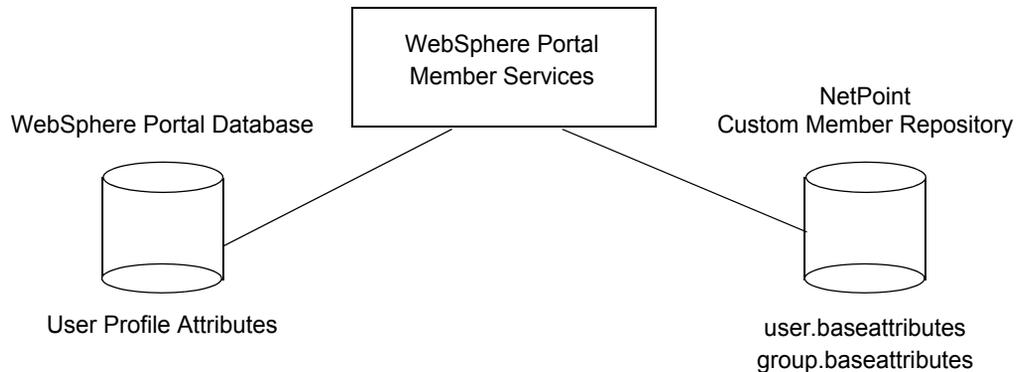
## NetPoint Custom Member Repository

The NetPoint Custom Member Repository (CMR) is available with the NetPoint Connector for WebSphere. The NetPoint CMR is an instance of a Member Services component that connects the WebSphere Portal Server to the NetPoint COREid system users and groups.

The CMR is a custom user data store that implements the IBM WebSphere MemberRepository interface. As shown in Figure 8, the NetPoint CMR stores user and group base attributes. The NetPoint CMR is used by the WebSphere Portal Server to make queries like `getAttributes` for a user for personalization, `getGroupMemberships`, search users by attribute, and similar functions. User

profile information is in the Portal database and authentication information is available through the Administration Console and LDAP.

**Figure 8** Member Services, WebSphere Portal Database, and the NetPoint CMR



The NetPoint CMR supports *only read* operations, not create, modify, or delete operations. The CMR is an extension of the custom user registry (CUR) and requires the Portal Server.

Two configuration files used to control WebSphere Portal Member Services come into play with the NetPoint CMR: `wms.xml` and `um.properties`. These are usually created during the portal installation.

- **wms.xml**—The `wms.xml` file configures the data sources used by the Member Services. The `ProfileDataStorage` parameter is set during portal installation. This parameter defines which user repository the Member Services will use. You will need to modify the `ProfileDataStorage` and `AuthenticationMode` values to enable use of the NetPoint CMR.
- **um.properties**—The `um.properties` file provides a mapping layer between the portal and Member Services. This file includes a comma-separated list of attribute names that will be passed to Member Services requests and several multi-valued properties. This file may need to be configured for the user

attributes for personalization. All user.baseattributes and group.baseattributes values will be searchable in the NetPoint CMR. For example:

```
user.baseattributes=cn,uid,cn,logonId,logonPasswordVerify,logonPassword ...
```

```
group.baseattributes=cn,uniqueOwnerIdentifier,membergroups,groupmembers,memberGroupName,memberGroupType,distinguishedName
```

All other attributes will go to the Portal database.

During startup, only the attributes identified in the getuser.minimum.attributes parameter are retrieved. For example:

```
getuser.minimum.attributes=cn,genUserid,cn,givenName,sn,mail
```

---

**Note:** All attributes in the getuser.minimum.attributes list must have correct attribute access control set in the NetPoint User Manager and Group Manager for the Administrator, and all need to be in one of the NetPoint User Manager configuration panels. For example, if the Portal Server needs the givenName attribute, one of the NetPoint panels needs First Name. In NetPoint, the givenName attribute is mapped to Display Name, First Name.

---

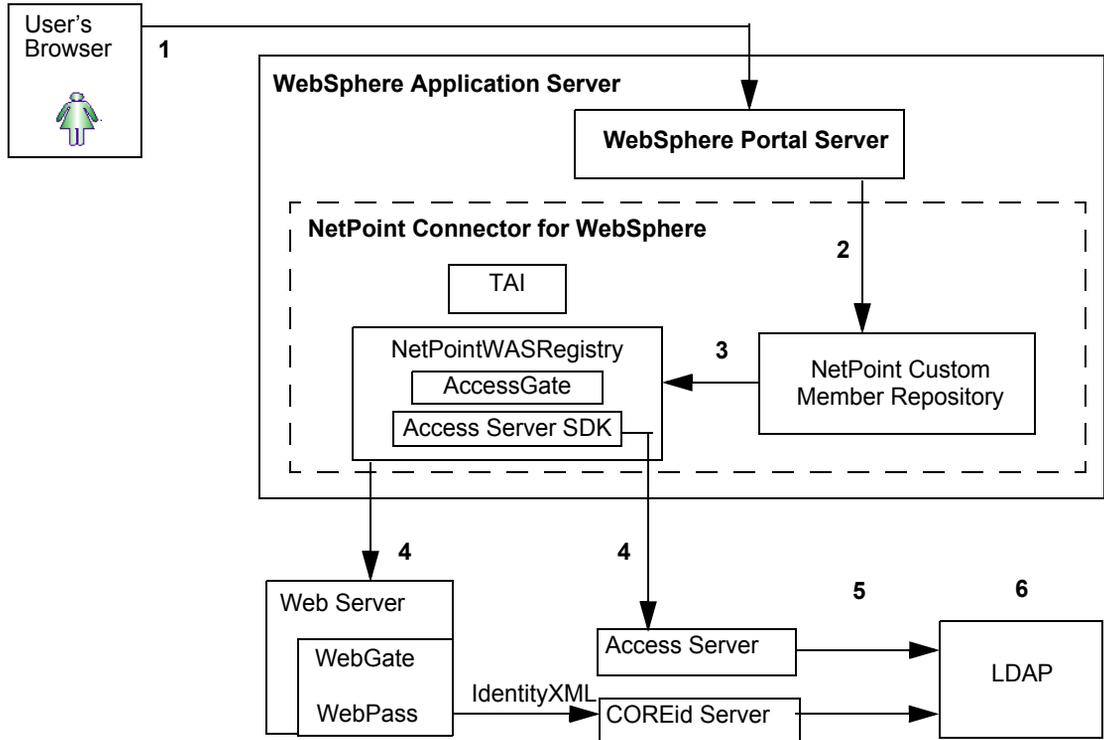
To see the LDAP to NetPoint mapping, you can select the desired attribute and view the corresponding Display Name in the COREid System Console > User Manager Configuration > Configure Tab > *Link* > Modify Attributes page. See the *NetPoint 7.0 Administration Guide Volume 1* for details.

## Integration Scenario with the NetPoint CMR

During login, the user is authenticated as depicted in “Integration Architecture” on page 184. Without the NetPoint CMR, the WebSphere Portal Server must communicate directly with the LDAP directory server to obtain user, group, and personalization information. With the NetPoint CMR, communication between the WebSphere Portal Server and the directory server can be eliminated. The NetPoint CMR performs read operations through the NetPointWASRegistry with the directory server.

Figure 9 shows the interaction between the WebSphere Portal Server, NetPoint CMR, and LDAP directory server during the login authorization process. This follows processes described in “Integration Architecture” on page 184.

**Figure 9** WebSphere Portal Server and the NetPoint Custom Member Repository



**Process overview: Authorization with the NetPoint CMR**

1. After authentication, a user requests access to a portlet through the WebSphere Portal Server.
2. The Portal Server forwards the request to the NetPoint CMR.
3. The NetPoint CMR forwards the request to the NetPointWASRegistry.
4. The NetPointWASRegistry sends an IdentityXML call to the COREid Server or uses the Access Server SDK to contact the Access Server through WebPass or WebGate, depending upon the required method.

For instance, the Access Server SDK uses the checkPassword method while IdentityXML uses all other methods:

- findByAttribute to search users by attribute
- get
- getGroupMemberIdentifiers
- getMemberAttributes
- getMemberships

- IsAttributeSupported
  - searchMembers
5. The COREid Server (or Access Server) communicates with the LDAP directory server.
  6. The directory server returns information.

## Setting up the WebSphere Portal for use with NetPoint

To integrate the NetPoint Connector for WebSphere with the Portal Server, all *must* share the same NetPoint installation to manage user and group information. This enables users and groups added through NetPoint to be immediately visible in the portal.

The portal must be configured to support *both* Database and LDAP directory modes. This enables the portal to search the LDAP directory for user profile attributes, then search for any additional profile attributes in the RDBMS database.

The Portal Server uses the WebSphere Application Server security framework for authentication. For this reason, you need to install the WebSphere Application Server with LDAP security, then change security to the CustomRegistry in the WAS Administration Console and specify an administrative user for the NetPointWASRegistry. This identifies the NetPointWASRegistry to authenticate and authorize portal users with NetPoint security policies.

When the NetPoint CMR is part of your environment, additional steps are needed to complete the implementation.

---

**Note:** Only the WebSphere Application Server (WAS) 4.x may be supported with the NetPoint 6.1.1.7 Custom Member Repository for the WebSphere Portal Server. For more information, see “Supported Versions and Platforms” on page 189.

---

### Task overview: Integrating NetPoint and the Portal Server

1. Define specifications for all NetPoint Integrations with the Portal Server, as discussed in “Integrating NetPoint and the Portal v4” on page 240.
2. Complete configuration of the NetPoint CMR if this is part of your environment, as discussed in “Configuring the NetPoint CMR” on page 241.

## Integrating NetPoint and the Portal v4

Some steps may refer to other topics in this chapter for more information. However, they are included here for completeness and include considerations and specifications peculiar to integration with NetPoint.

### To integrate NetPoint with the WebSphere Portal Server

1. Complete tasks in “Preparing to Install the NetPoint Connector” on page 190 *and* ensure that you:
  - a) Set security to *LDAP* when you install the WebSphere Application Server v4.x.
  - b) Specify *Database and LDAP Directory mode* as the authentication mode for Portal Member Services when you install the WebSphere Portal Server, then deploy all portlets.
  - c) Install and configure NetPoint, as discussed in “Preparing Your Environment” on page 191.
2. Install the NetPoint Connector and configure the NetWASRegistry and TAI components, as discussed in “Installing the NetPoint Connector for WebSphere” on page 206.
3. Complete NetPoint Connector setup and testing, as discussed in “Completing NetPoint Connector Setup” on page 217.
4. Configure the Application Server, as described in “Configuring WebSphere Application Server v4” on page 221 and in the Administration Console, change security to the CustomRegistry as shown below:

**From**—LDAP  
**To**—CustomRegistry
5. In the Admin Role tab, specify a user for the administrative role for the NetPointWASRegistry, as described in “Specifying an Administrative Role” on page 224.
6. Complete configuration for the NetPoint CMR, if this is part of your environment, as discussed next.

## Configuring the NetPoint CMR

You must complete the following steps to implement the NetPoint CMR with the WebSphere Portal Server.

### To complete configuration for the NetPoint CMR

1. Copy the three files indicated below regardless of your platform.
  - a) Copy the NetPointCMR.jar file from *NPCWS\_install\_dir/oblix/lib* to *WAS\_install\_dir/lib/app*.
  - b) Copy the NetPointWASRegistry.jar file from *NPCWS\_install\_dir/oblix/lib* to *WAS\_install\_dir/lib*.
  - c) Copy the jobaccess.jar file from *NPCWS\_install\_dir/oblix/lib* to *WAS\_install\_dir/lib*.

where *NPCWS\_install\_dir* is the directory where the NetPoint Connector for WebSphere is installed and *WAS\_install\_dir* is the directory where the WebSphere Application Server is installed.

NetPoint requires the Admin username and password to make IdentityXML calls to the WebPass. For details about administrator rights, see “Configuring the COREid Server for WAS Search Methods and the NetPointWASRegistry Admin” on page 193.

2. Ensure that the following Admin credentials are set in *clear text* in the NetPointWASRegistry.properties file:

```
OB_AdminUserName=admin
OB_AdminUserCreds=password
```

where the OB\_AdminUserName value is the userid of the Portal Server administrator who is a NetPoint Master Identity Administrator or NetPoint Administrator.

This is required for the CMR. The Admin credentials should be set in clear text. The NetPointWASRegistry reads the password, encrypts it, and re-writes the properties file with the encrypted password. The encryptor can be executed by running the registryTester program, as well as from WebSphere. To assist you with adding these parameters, see the NetPointWASRegistryProperties.sample file, which includes comments. See also, “NetPointWASRegistry.properties” on page 283.

---

**Note:** NetPointWASRegistry.properties file formatting is lost when NetPoint re-writes the file with the encrypted password. You may want to save a copy of the NetPointWASRegistry.properties.

---

3. Copy the wms.xml file and store the original as a backup before you continue.

**Copy**—*WAS\_install\_dir/lib/app/xml/wms.xml*

**To**—*WAS\_install\_dir/lib/app/xml/wms.xml.original*

4. In the wms.xml file MemberSubSystem section, change the AuthenticationMode and ProfileDataStorage values to read as follows:

```
<MemberSubSystem
    name="Member SubSystem"
    AuthenticationMode="OTHER"
    ProfileDataStorage="PLUGIN"
...>
```

The original value for both AuthenticationMode and ProfileDataStorage is LDAP. However, to operate with the NetPoint CMR, the value must be OTHER and PLUGIN, respectively. <Directory> information may remain in this file even though it will be ignored.

Next you must edit the Plugin section to point to the appropriate NetPoint files. In the step below, changing the pluginImplClass and pluginEnvProperty parameters is required. Be certain to use the form shown in step 5.

5. In the wms.xml file Plugin section, edit the pluginImplClass and pluginEnvProperty values using the form shown below:

```
<Plugin
pluginImplClass="com.oblix.registry.NetPointMemberRepositoryImpl"
EntryFileName=""
orgEntitySupported="0">
<pluginEnvProperty name="NetPointWASRegistry.properties">
c:/NPCWS_install_dir/oblix/config/NetPointWASRegistry.properties
</pluginEnvProperty>
<pluginEnvProperty name="um.properties">
/config/um.properties</ "missing tag here, complete as </pluginEnvProperty">
</Plugin>
```

6. Copy the um.properties file and store the original as a backup before you continue.

**Copy**—*WAS\_install\_dir/lib/app/config/um.properties*

**To**—*WAS\_install\_dir/lib/app/config/um.properties.original*

The values in um.properties are case *insensitive*, *not* case sensitive.

7. In the um.properties file:

- a) Ensure that the following filters are correct for your environment and edit the values if needed:

**user.baseattributes**—

cn,uid,cn,logonId,logonPasswordVerify,logonPassword ...

**group.baseattributes**—

cn,uniqueOwnerIdentifier,membergroups,groupmembers,memberGroupName,memberGroupType,distinguishedName

- b) Ensure that the `getuser.minimum.attributes` values include all attributes for the user that the CMR will retrieve:

**getuser.minimum.attributes**—cn,genUserid,cn,givenName,sn,mail

- c) Add the following in the `um.properties` file, which the CMR uses to retrieve Group information for the Portal:

**getgroup.minimum.attributes** =uniqueMember, cn

8. Ensure the following:

- Ensure that all `getuser.minimum.attributes` have the correct attribute access control for the Administrator.
- Ensure that all `getgroup.minimum.attributes` are in one of the NetPoint panels: COREid System Console > Group Manager Configuration > Configure Tab > Link > View Object Profile > Configure Group Profile Panels

See the *NetPoint 7.0 Administration Guide Volume 1* for more information.

9. Restart the Portal Server.

You can enable logging for the NetPointWASRegistry using parameters in the `NetPointWASRegistry.properties` file. Log messages are directed to different locations depending upon your configuration:

- **Without the CMR**—Log messages for the NetPointWASRegistry are directed to the file named in the `OB_LogFileName` parameter.
- **With the CMR**—Log messages for NetPointWASRegistry are directed to the file named in the `OB_LogFileName` parameter, and log messages for the CMR are directed to the `WPS_install_dir/log/appserver-out.log` file.

For details about enabling logging, see “NetPointWASRegistry.properties” on page 283.

10. Log in to `http://host:port/wps/portal` and search for users and groups listed in the portal database to ensure that the integration is successful.

where *host:port* is the name and port number of the host machine and *portal* is the name of your portal page.

## Managing Users and Groups

Portal Administrators can use the NetPoint COREid System to perform user and group management tasks such as adding or deleting users and groups, modifying user profiles and attributes.

You can add and delete static groups and user membership in groups through the NetPoint COREid System. The information that you update in NetPoint is immediately reflected in the WebSphere Portal. After you create users and groups in the COREid System, you can search for them in the WebSphere Portal.

- To use NetPoint user and group management functionality, ensure that you do *not* create users and groups in the WebSphere Portal. Instead, create and modify users and groups in the NetPoint COREid System.
- To recognize group membership, NetPoint requires the dynamic group to be expanded.

---

**Note:** With the CMR, NetPoint does not require the dynamic group to be expanded to recognize group membership.

---

See the *NetPoint 7.0 Administration Guide Volume 1* for more information on managing users and groups.

## Modifying User Profiles and Attributes

When users modify their profile through the NetPoint COREid System, the modifications are immediately visible in the WebSphere Portal. This ensures that the most current user information is used when portal developers personalize user pages.

You can map additional attributes to a user's profile if necessary. See the WebSphere Portal documentation for information on mapping attributes.

---

**Note:** The NetPoint CMR has no impact on this.

---

## Password Management

Because the portal uses NetPoint SSO, users are subject to the NetPoint password policies during authentication.

---

**Important:** To implement the NetPoint password management feature, turn off the portal's password management functionality. The NetPoint CMR has no impact on this.

---

The NetPoint password management feature provides functionality such as defining password policies, resetting passwords, expiration notification, and challenge phrases for lost passwords.

See the *NetPoint 7.0 Administration Guide Volume 1* for more information on password policies.

## Access Control for the WebSphere Portal

Portal administrators use the portal's access control functionality to grant access to portlets. From the WebSphere Portal, administrators can search for NetPoint-managed users and groups to whom they want to grant portal administration privileges as well as portlet access control.

## Configuring Single Sign-On Functions for the Portal

Configuring SSO between NetPoint and the WebSphere portal enables the WebSphere portal to utilize the ObSSOCookie and enable NetPoint Connector for WebSphere to authenticate NetPoint users.

Configuring SSO logout for the WebSphere Portal Server ensures that when a user logs out of a NetPoint-protected WebSphere resource, both the LTPA token and the ObSSOCookie are killed. The user will not be able to access any other WebSphere resource or other NetPoint-protected resources without authenticating again.

- If you have configured the TAI for SSO between NetPoint and WebSphere, you must configure single sign-on logout for the WebSphere Portal Server.
- If you have not configured the TAI for SSO, users can use the portal's logout button to log out of all NetPoint-protected resources.

---

**Note:** The NetPoint CMR has no impact on this.

---

### To configure SSO for the WebSphere Portal

1. Install the WebGate plug-in for the Web server that you selected when you installed the WebSphere Portal.
2. In the NetPoint Access Manager, define the URL that you want to protect.

For example:

*myhost:port/wps/myportal/*

where *myhost:port* represents the hostname and port number within a specific URL.

WebGate prompts for authentication when users attempt to log in to this URL. Be sure to protect / if you want WebGate to prompt for authentication when the

user gets to the root of the WebSphere Portal. You can also add other authorization rules, if needed.

---

**Note:** To protect resources, it is recommended that you use a form-based authentication scheme. If you use the basic authentication scheme, be sure to set the Challenge Redirect field to another WebGate to ensure that the ObSSOCookie gets set. See the *NetPoint 7.0 Administration Guide Volume 2* for more information on authentication schemes.

---

## To configure SSO logout

1. Create a NetPoint policy with a Form Over LDAP type of authentication scheme to protect the portal URL.

See the *NetPoint 7.0 Administration Guide Volume 2* for information on creating NetPoint policies.

2. Create a custom logout page using HTML, JSP, or CGI protocol.

The default NetPoint logout page, `logout.html`, is located in:

`WebGate_install_dir\access\oblix\apps\common\bin`, where `WebGate_install_dir` is the directory where the WebGate is installed.

3. Save the logout page in the document root of the Web server on which the WebGate that protects WebSphere is installed.

For example:

`http://foobar/myportal/logout.html`

---

**Note:** Ensure that the name of the logout page contains the string “logout.”.

---

4. Protect the logout page with a NetPoint policy.

See the *NetPoint 7.0 Administration Guide Volume 2* for information on protecting resources with a NetPoint policy.

5. Open the file `ConfigServices.properties` located in `WAS_install_dir\lib\app\config\services`.

where `WAS_install_dir` is the directory where you installed the WebSphere Application Server. For example: `C:\IBM\WebSphere\AppServer`.

6. Add the following two parameters in `ConfigServices.properties` file:

- `redirect.logout = true`
- `redirect.logout.url =` The path to the logout page.

For example:

`http://foobar/myportal/logout.html`

7. Restart the WAS, which will restart the WebSphere Portal Server.

# Configuring WebSphere Application Server v5

The following sections describe the integration of NetPoint with the WebSphere Application Server v5.0 or 5.1. If you are using the NetPoint 7.0 Connector, skip to “Configuring WebSphere Application Server v4” on page 221.

Before you begin, see “Supported Versions and Platforms” on page 189 for complete details about version support.

## Task overview: Integrating NetPoint with WAS v5

1. “Enabling the NetPointWASRegistry in WAS v5” on page 247
2. “Testing the NetPointWASRegistry for WebSphere v5” on page 251
3. “Configuring the TAI for WebSphere v5” on page 251

## Enabling the NetPointWASRegistry in WAS v5

Once you have installed the products and tested them to be sure they are communicating, you can enable the NetPointWASRegistry. Enabling the registry allows NetPoint to be used as the authentication source for the WebSphere Application Server. .

The NetPointWASRegistry works with the User Registry in WebSphere 5.

### To enable the NetPointWASRegistry in WAS 5

1. Ensure that WebGate protecting the WebPass has IPValidation set to False.
2. Copy the security.xml file located in the directory below:

*WAS\_install\_dir/config/cells/serverName*

---

**Note:** It is important to create a backup copy of security.xml whenever you make a change to the configuration. If there are errors in the new configuration, you can always restore the previous version of the security.xml file.

---

3. Start the WebSphere Admin Service.
4. Log in as the Identity Administrator into the WebSphere Administrative Console.

---

**Note:** Any NetPoint user added to the role “Admin” must be in a group called “Admin” or “Administrator”. Otherwise, this user may not be able to login to WebSphere Administrative Console. Other roles, like “monitor” have no restrictions. So any NetPoint user added to these roles in the WAS 5 Administrative Console, can log into the WAS 5 Administrative Console.

---

5. Navigate to Security > User Registries > Custom Properties and enter the following:
  - Identity Administrator user ID
  - Server User Password
  - CustomRegistry Classname
  - NetPoint Admin ID
  - User's password
  - NetPoint Classname: com.oblix.registry.NetPointWASRegistry

**Custom User Registry**

A custom user registry that implements the com.ibm.websphere.security.UserRegistry interface. For backward compatibility, a custom user registry that implemented the com.ibm.websphere.security.CustomRegistry interface are also supported. When security is enabled and these properties are changed, please go to the GlobalSecurity panel and click Apply or OK to validate the changes. [?](#)

**Configuration**

**General Properties**

Server User ID	* admin
Server User Password	* *****
Custom Registry Classname	* com.oblix.registry.NetPointWASRegistry
Ignore Case	<input type="checkbox"/>

Apply OK Reset Cancel

**Additional Properties**

[Custom Properties](#) A set of arbitrary user registry configuration properties whose names are specific to a given type of pluggable registry.

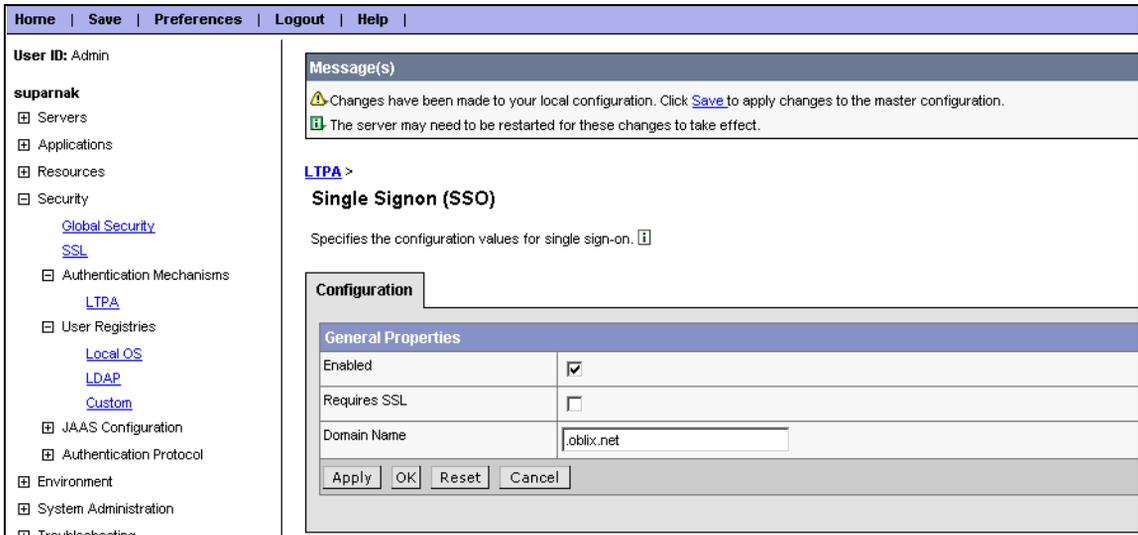
6. Under Additional Properties in the Configuration tab, click Custom Properties, click New, and enter the following:
  - Name: NetPointWASRegistry.properties
  - Value: C:\NPCWS\_install\_dir\oblix\config\NetPointWASRegistry.properties.
  - Description: Property file for NetPoint User Registry.

- Select the Required checkbox, click OK, and save your changes.

The screenshot shows the NetPoint WAS Registry configuration interface. The left navigation pane is expanded to 'Authentication Mechanisms' > 'Custom'. The main area displays the 'NetPointWASRegistry.properties' configuration tab. The 'Configuration' sub-tab is active, showing the 'General Properties' section. The 'Name' field is 'NetPointWASRegistry.properties', the 'Value' field is 'C:\NP65Beta\NetPointWASRegistry\o', the 'Description' is 'Property file for NetPoint User Regist', and the 'Required' checkbox is checked. The 'Validation Expression' field is empty. Buttons for 'Apply', 'OK', 'Reset', and 'Cancel' are visible at the bottom of the configuration area.

7. In the navigation pane on the left, click Authentication Mechanism > LTPA.
8. In the Configuration tab, specify a password.
9. Click OK.
10. Navigate back to LTPA (in the navigation pane on the left, click Authentication Mechanism > LTPA) and do the following:
  - Click Single Signon (SSO).
  - Click the Enabled box.
  - Enter the domain name. For example, .oblix.com.

- Click OK.



11. In the navigation pane on the left, click Security > Global Security and change the following:
  - a) Set the Active Authentication Mechanism to LTPA.
  - b) Set the Active User Registry to Custom.
  - c) Click OK.
  - d) Click the Enabled box.
  - e) Click Apply to test the configuration.
 

If the information is correct, a message confirming the changes is displayed at the top. Correct any errors that are displayed.
  - f) Click Save.
  - g) Click Logout and close the browser window.
  - h) Stop the WebSphere Application Service.

If you get a message stating that the Service could not be stopped, switch to the Task Manager and ensure that there are no java processes running.

12. Start the WebSphere Admin Service.

## Testing the NetPointWASRegistry for WebSphere v5

Next, verify the NetPointWASRegistry is configured correctly.

### To test the NetPointWASRegistry configuration

1. Access the Snoop Servlet sample running on the default server at:

`http://hostname:port/snoop`

or

`http://hostname:web_server_plug-in port/snoop`

2. When challenged by WebSphere, enter a username and password that is valid in NetPoint.

By default, any authenticated user should be allowed access.

3. Launch the WebSphere Administrative Console and login as the user specified in Security > User Registries > Custom Properties.

If the configuration is correct, you will be able to login successfully.

4. Set access control for the WebSphere Administrative Console by specifying the users and groups and the roles to which they belong.

See WebSphere 5 documentation for more information.

5. Note the .xml files that are modified to support Admin Console access and click Save.

After you have installed NetPoint Connector for WebSphere, you must configure the NetPointWASRegistry and TAI. The NetPointWASRegistry is used for authentication and the TAI is used for NetPoint single sign-on.

## Configuring the TAI for WebSphere v5

You configure the TAI to enable single sign-on between NetPoint and WAS, as well as between NetPoint and the WebSphere Portal Server. For WebSphere 5.0 or 5.1, you must install `webgate.properties`, add the TAI, and then add custom properties.

---

**Note:** For optional details, see “Implementation Notes for the TAI” on page 292.

---

## To install and configure TAI for WAS 5

1. Copy the configuration file named `webgate.properties` (see Table 12 for parameters):

**From:** `NPCWS_install_dir/oblix/config`

**To:** `WAS_install_dir/properties` folder in the WebSphere installation properties directory.

where `NPCWS_install_dir` is the directory where NetPoint Connector for WebSphere is installed, and `WAS_install_dir` is the directory where the WebSphere Application Server is installed.

This file contains configuration information that WebSphere will use to connect to the AccessGate.

2. In the WebSphere installation properties directory, modify the parameter values of the `webgate.properties` file (Table 12) as follows:

`OB_InstallDir = NPCWS_install_dir`

where `NPCWS_install_dir` is the directory where NetPoint Connector for WebSphere is installed. For example:

`C:\NetPoint\NetPointWASRegistry`

If WebGate is installed on a proxy server that is used as a front end server to direct all user requests to Web servers that interface with WebSphere Application Servers, then specify the following parameter values:

- `OB_IsProxyEnabled=true`
- `OB_hostnames = serverName`  
where `serverName` is the name of the proxy server.
- `OB_ports = portNumber`  
where `portNumber` is the port number of the proxy server.

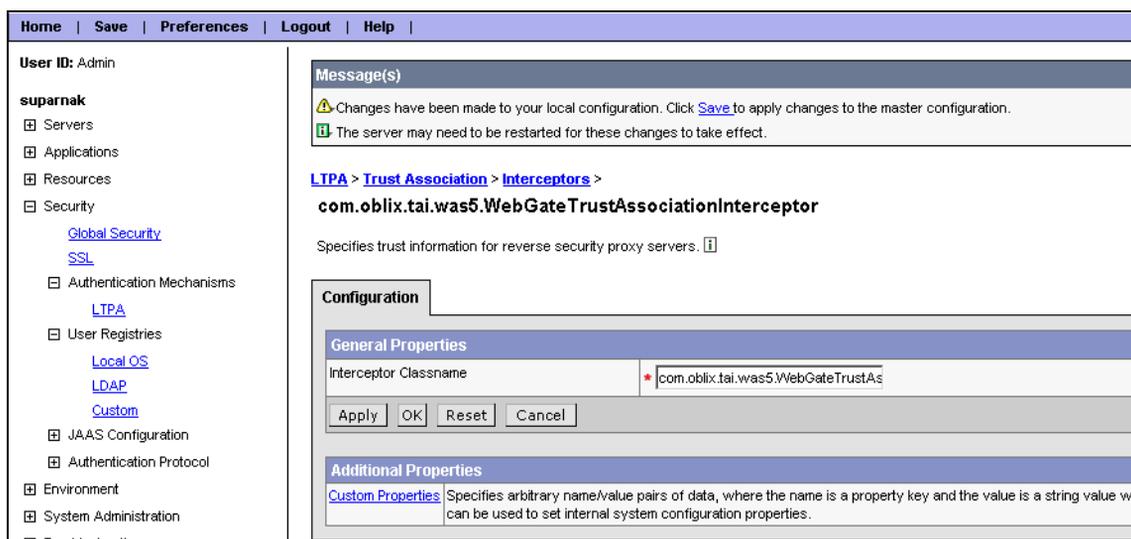
---

**Note:** If you used a resource other than Authen, you must specify the resource name in the `webgate.properties` file.

---

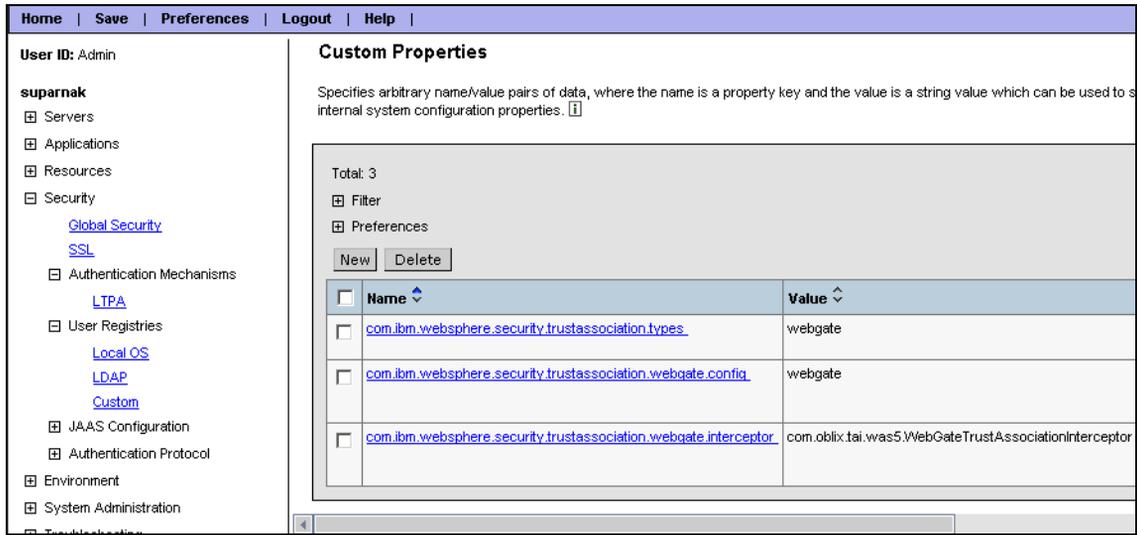
3. Launch the WebSphere Administrative Console
4. In the navigation pane on the left, click Authentication Mechanisms > LTPA.
5. Under Additional Properties, click Trust Association > Interceptors > New.
6. In the Name field, enter Oblix TAI Interceptor.

7. In the Interceptor classname field, enter `com.oblix.tai.was5.WebGateTrustAssociationInterceptor`



8. Click OK.
9. Add the following three properties:
  - a) **Name**—`com.ibm.websphere.security.trustassociation.types`
    - **Value**—`webgate`
    - **Description**—NetPoint TAI property
    - Select the Required checkbox.
    - Click OK.
  - b) **Name**—`com.ibm.websphere.security.trustassociation.webgate.config`
    - **Value**—`webgate`
    - **Description**—Name of the NetPoint TAI properties file located in `WAS_install_dir/properties` directory.
    - Select the Required checkbox.
    - Click OK.
  - c) **Name**—`com.ibm.websphere.security.trustassociation.webgate.interceptor`
    - **Value**—`com.oblix.tai.was5.WebGateTrustAssociationInterceptor`
    - **Description**—NetPoint TAI class for WebSphere 5
    - Select the Required checkbox.

- Click OK.



10. Click Interceptors at the top of the page and then click Save.
11. Navigate to:  
`WAS_install_dir/config/cells/serverName_dir`  
 where `serverName_dir` is the directory where the server is installed.
12. Make a backup copy of the security.xml file.
13. In the WebSphere Administrative Console, navigate to LTPA > Trust Association > Interceptors.
14. Select the WebSeal Interceptor and click Delete.
15. Click Trust Association and click the Enabled check box.
16. Click Apply and then click Save.
17. Logout of the WebSphere Administrative Console and close the browser.
18. Shut down the Websphere Admin Server.  
 If you get an error message, go to Task Manager and ensure that the java process is not running.
19. Restart the WebSphere Admin Service.  
 If the Service does not start, verify that the properties are set correctly in the security.xml file and that the webgate.properties file is in the correct location.
20. Create a NetPoint policy to protect WebSphere resources as described in “To install and configure TAI for WAS 4” on page 227.
21. Verify that the TAI is working as detailed below.

## Testing the TAI for WAS v5

### To test the TAI

1. Restart WAS.
2. For additional information on testing, see the discussion “Testing NetPoint Single Sign-on for WAS v4.x” on page 231, as this is the same for both WAS v4.x and WASv5.x.

## Enabling Logging for TAI for WAS v5

You can enable logging for the TAI from the WebSphere Administrative Console. The procedure to enable logging varies depending on the WAS version that you use.

### To enable logging for TAI for WAS 5

1. Launch the WebSphere Administrative Console.
2. Navigate to Troubleshooting > Logs and Trace
3. Select your Server.
4. Select Diagnostic Trace.
5. Modify the Trace specification.
6. Select the Components tab.
7. Enable debug logging for `com.oblix.tai.was5.WebGateTrustAssociationInterceptor`.
8. See “Testing NetPointRegistry for WAS v4, v5, and v6” on page 224.
9. Integrate NetPoint with the WAS Portal v5, if desired.

## Integrating NetPoint and WebSphere Portal v5

A portal provides a single point of access to enterprise data and applications, presenting a unified and personalized view of that information to employees, customers, and business partners.

The WebSphere Portal Server runs on top of the WAS and uses the WAS security infrastructure to enforce access control. Integrating NetPoint with the WebSphere Portal provides the following NetPoint functionality for the portal:

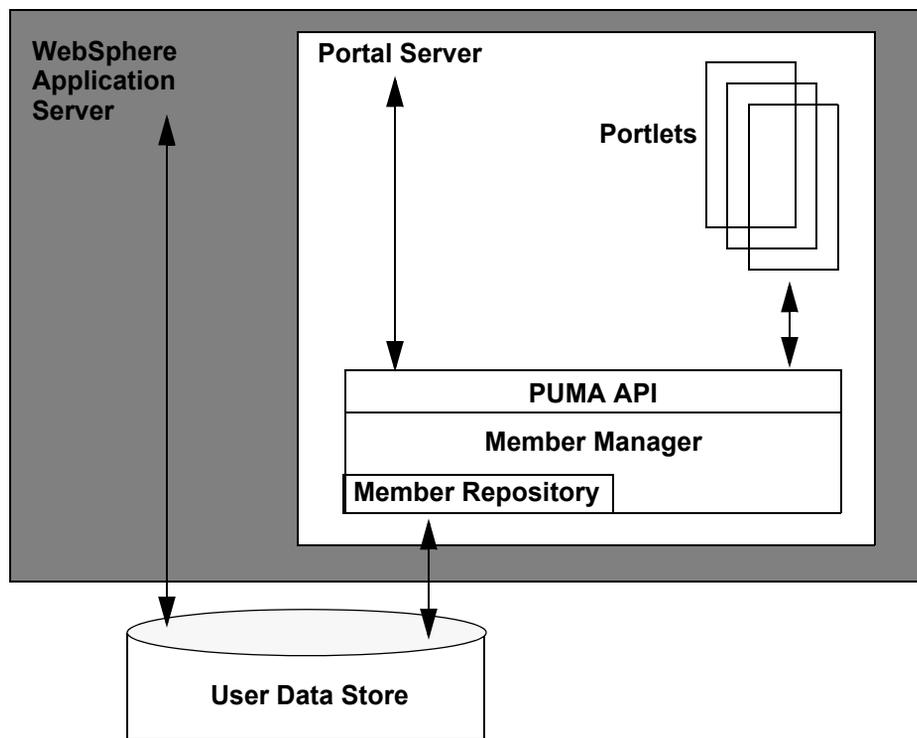
- User and group management
- Password management

- SSO to the portal
- Unified logout between NetPoint, WAS, and the WebSphere Portal

The WebSphere Portal V5 uses the following component to manage user and group information.

**Member Manager**—Member Manager presents a Java object view of Users and Groups to WebSphere Portal, including all portlets installed on WebSphere Portal. Member Manager (as accessed through PUMA) is the abstraction interface that WebSphere Portal V5.0 uses to access user and group information. This includes the user accounts, which tell Portal that the user exists, any user groups within which the user might be a member, and attributes about the users.

The following figure shows the architecture of the Member Manager.



See the WebSphere Portal v5 documentation for more information on the portal and related components.

**NetPoint Custom Member Repository**—The NetPoint Custom Member Repository (CMR) is available with the NetPoint 7 Connectors for WebSphere, as described in “Supported Versions and Platforms” on page 189. The NetPoint CMR is an instance of a Member Manager component that connects the WebSphere Portal Server to the NetPoint COREid system users and groups.

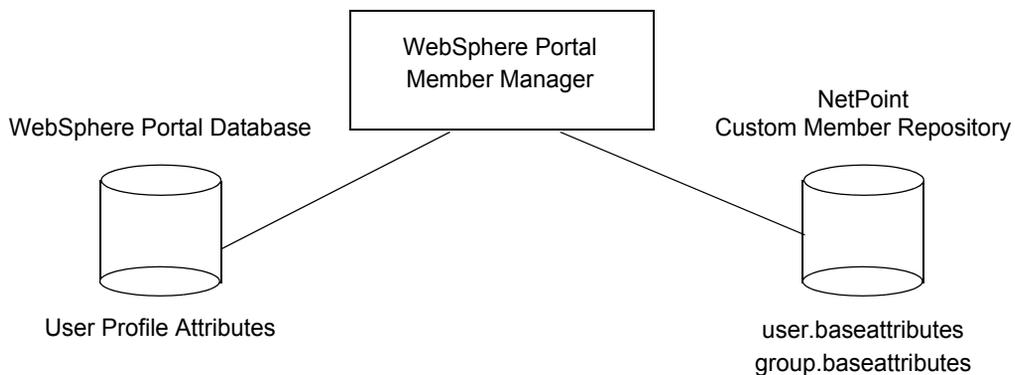
The CMR is a custom user data store that implements the IBM WebSphere MemberRepository interface. As shown in next figure, the NetPoint CMR stores user and group base attributes in user data store. The NetPoint CMR is used by the WebSphere Portal Server to make queries like `getAttributes` for a user for personalization, `getGroupMemberships`, search users by attribute, and similar functions. User profile information is in the Portal database and authentication information is available through the Administration Console and LDAP.

---

**Important:** Group MemberShip functionality is not currently supported. As a result, with Nested Groups, if you check for inner group membership it won't display its parent group details.

---

**Figure 10** Member Manager, WebSphere Portal Database, and the NetPoint CMR



The NetPoint CMR supports only read operations, not create, modify, or delete operations. The CMR is an extension of the custom user registry (CUR) and requires the Portal Server.

The configuration files used to control WebSphere Portal Member Manager come into play with the NetPoint CMR are explained below, which are usually created during the portal installation:

**wmm.xml**—Top level Member Manager configuration. Lists and configures the various MemberRepository implementations used by Member Manager. Most other Member Manager configuration files are pointed to from this file. The CMR details should be configured in this file.

**PumaService.properties**—This is the configuration file for the PUMA API, which in WebSphere Portal V5.0.x is a mapping layer between WebSphere Portal and Member Manager. This is not a Member Manager configuration file, but because PUMA is part of the "user stack" in Portal, this configuration file is important.

This file includes a comma-separated list of attribute names that will be passed to Member Manager requests and several multi-valued properties. This file may need to be configured for the user attributes for personalization. All `user.base.attributes` and `group.base.attributes` values will be searchable in the NetPoint CMR. For example:

```
user.base.attributes=cn,uid,cn,logonId,logonPasswordVerify,logonPassword
...
group.base.attributes=cn,uniqueOwnerIdentifier,membergroups,groupmembers,memberGroupName,memberGroupType,distinguishedName
```

All other attributes go to the Portal database.

During startup, only the attributes identified in the `user.minimum.attributes` parameter are retrieved. For example:

```
user.minimum.attributes=cn,genUserid,cn,givenName,sn,mail
```

---

**Note:** All attributes in the `user.minimum.attributes` list must have correct attribute access control set in the NetPoint User Manager and Group Manager for the Administrator, and all need to be in one of the NetPoint User Manager configuration panels. For example, if the Portal Server needs the `givenName` attribute, one of the NetPoint panels needs First Name. In NetPoint, the `givenName` attribute is mapped to Display Name, First Name.

---

To see the LDAP to NetPoint mapping, you can select the desired attribute and view the corresponding Display Name in the COREid System Console > User Manager Configuration > Configure Tab > Link > Modify Attributes page.

See the *NetPoint 7.0 Administration Guide Volume 1* for details.

**wpconfig.properties**—This is WebSphere Portal configuration file. The portal user/group administrator details needs to be set in this file. This file is present in `WPS_install_dir/config` folder. Where `WPS_install_dir` is the Portal Server home directory.

**VaultService.properties**—This file is located in `WPS_install_dir/shared/app/config/services` folder. This configuration file is used to specify Vault Adapter Implementations. You have to set correct system admin credential DN in this file.

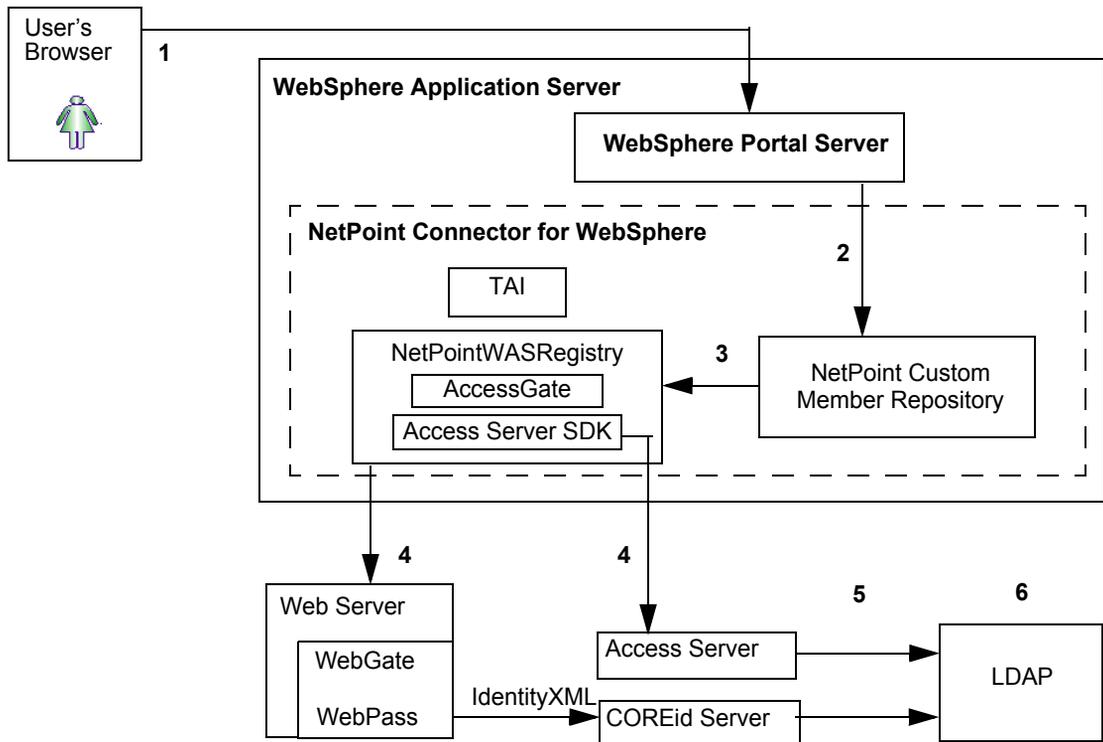
## About Integration with the NetPoint CMR

During login, the user is authenticated as depicted in “Scenario 1: Use of NetPointWASRegistry” on page 184.

Without the NetPoint CMR, the WebSphere Portal Server must communicate directly with the LDAP directory server to obtain user, group, and personalization information. With the NetPoint CMR, communication between the WebSphere Portal Server and the directory server can be eliminated. The NetPoint CMR performs read operations through the NetPointWASRegistry with the directory server.

The next figure shows the interaction between the WebSphere Portal Server, NetPoint CMR, and LDAP directory server during the login authorization process. This follows processes described in “Scenario 1: Use of NetPointWASRegistry” on page 184.

**Figure 11** WebSphere Portal Server and the NetPoint Custom Member Repository



### Process overview: Authorization with the NetPoint CMR

1. After authentication, a user requests access to a portlet through the WebSphere Portal Server.
2. The Portal Server forwards the request to the NetPoint CMR.

3. The NetPoint CMR forwards the request to the NetPointWASRegistry.
4. The NetPointWASRegistry sends an IdentityXML call to the COREid Server or uses the Access Server SDK to contact the Access Server through WebPass or WebGate, depending upon the required method.

For instance, the Access Server SDK uses the checkPassword method while IdentityXML uses all other methods:

- findByAttribute to search users by attribute
  - getMember
  - getGroupMemberIdentifiers
  - getMembers
  - search etc.
5. The COREid Server (or Access Server) communicates with the LDAP directory server.
  6. The directory server returns information.

## Setting up the WebSphere Portal v5.0.2 for NetPoint

Integrating the WebSphere Portal v5 with NetPoint involves a series of installation and configuration tasks.

### To integrate the WebSphere Portal with NetPoint

1. Complete tasks in “Preparing to Install the NetPoint Connector” on page 190:
  - a) Install the WebSphere Application Server with appropriate Fix Pack as described in the WebSphere documentation to correct the following issues.

For example:

**Issue:** If user dn contents intermediate spaces after comma, like

Cn=PortalUser, o=company, c=us then access permissions to portlets was not working for such users.

IBM has released a Fix PQ93461 for this Portal User access permission problem.

**Action:** This fix pack needs to be applied. Please verify the Portal Server History Log to ensure all required fixes have been installed. Refer Portal Infocenter document for more information on this

- b) Install the WebSphere Portal with WebSphere Application Server security disabled.

---

**Note:** Both the Global and Java 2 security should be disabled while installing the Portal Server.

---

See the IBM WebSphere Portal Infocenter document for information on installation.

- c) Install and configure NetPoint, as discussed in “Preparing to Install the NetPoint Connector” on page 190.
2. Install the NetPoint Connector for WebSphere and configure the NetpointWASRegistry and TAI components, as described in “Installing the NetPoint Connector for WebSphere” on page 206.
3. Complete NetPoint Connector setup and testing, as discussed in “Completing NetPoint Connector Setup” on page 217.
4. In the WebSphere Administration Console, change security to custom registry. This specifies the NetPointWASRegistry, which establishes a connection between the WAS and NetPoint. The WAS uses the NetPointWASRegistry to authenticate and authorize portal users with NetPoint’s security policies.
5. Ensure that the following Admin credentials are set in clear text in the NetPointWASRegistry.properties file:

```
OB_AdminUserName=admin  
OB_AdminUserCreds=password
```

where the OB\_AdminUserName value is the userid of the Portal Server administrator who is a NetPoint Master Identity Administrator or NetPoint Administrator.

This is required for the CMR. The Admin credentials should be set in clear text. The NetPointWASRegistry reads the password, encrypts it, and re-writes the properties file with the encrypted password. The encryptor can be executed by running the registryTester program, as well as from WebSphere. To assist you with adding these parameters, see the NetPointWASRegistryProperties.sample file, which includes comments. See also, “NetPointWASRegistry.properties” on page 283.

---

**Note:** NetPointWASRegistry.properties file formatting is lost when NetPoint re-writes the file with the encrypted password. You may want to save a copy of the NetPointWASRegistry.properties.

---

6. Restart the WAS Server.
7. Ensure that the WebSphere Application server and Portal Server is stopped.

8. Make a back-up copy of the file below:

*WPS\_install\_dir*/config/wpconfig.properties file

where *WPS\_install\_dir* is the directory where WebSphere Portal Server is installed.

9. Edit the wpconfig.properties file and add the following:

PortalAdminId (Example: PortalAdminId= DN of wpsadministrator)

PortalAdminIdShort (Example: PortalAdminId= wpsadministrator)

PortalAdminPwd (Example: PortalAdminPwd=wpsadminpassword)

wasUserId (Example: wasUserId=wasadministrator )

wasPassword (Example: wasPassword=wasadminpassword)

PortalAdminGroupId (Example: PortalAdminGroupId= DN of PortalAdminGroupId)

PortalAdminGroupIdShort (Example: PortalAdminGroupIdShort= PortalAdminGroupId)

10. Restart the WebSphere Application server.

11. **Optional**—Turn PUMA traces on in Portal, which can be done by entering the following in the log.properties file.

For example:

*WPS\_install\_dir*\shared\app\config\log.properties

```
traceString=com.ibm.wps.services.puma.*=all=enabled:com.ibm.wps.puma.*=all=enabled:com.ibm.wps.command.puma.*=all=enable
```

12. Backup the file below:

*WPS\_install\_dir*/shared/app/wmm/wmm.xml file

13. Edit the file to make the changes for Netpoint CMR configuration as follows (and the lookaside flag should be set to “false”).

```
<supportedMemberTypes>
```

```
  <supportedMemberType name="Person"
```

```
    rdnAttrTypes="uid"
```

```
    defaultParentMember="o=company,c=us" //Root DN
```

```
    defaultProfileRepository="CNR"/>
```

```
  <supportedMemberType name="Group"
```

```
    rdnAttrTypes="cn"
```

```
    defaultParentMember="o=company,c=us"
```

```
    defaultProfileRepository="CNR"/> //Name of the CMR
```

```
information tag
```

```
</supportedMemberTypes>
```

```
<profileRepository name="NetpointCustomRepository" UUID="CNR"
```

```

description="This is Netpoint WMM custom MemberRepository
implementation." vendor="OBLIX!"
adapterClassName="com.oblix.registry.NetPointMemberRepositoryImpl_
v5"
//Name of the CMR implementation class
specVersion="1.0" adapterVersion="2.0"
configurationFile="customRepositoryAttributes.xml"
wmmGenerateExtId="true" supportGetPersonByAccountName="false"
supportDynamicAttributes="false" profileRepositoryForGroups="CNR"
enableTrace="true" PumaService.properties="WPSInstallDir/
shared/app/config/services/PumaService.properties" //Path of the
PUMA service file NetPointWASRegistry.properties="
NetpointWASConnInstallDir\oblix\config\NetPointWASRegistry.propert
ies">
//Path of the Netpoint WAS registry configuration file.

```

**Unix**—This path should be mentioned as “NetpointWASConnInstallDir/oblix/config/NetPointWASRegistry.properties”:

```

<readMemberType>
    <memberType name="Person" /> <memberType
name="Group" />
    //Only read access to CMR
</readMemberType>
<createMemberType>
    <!-- <memberType name="Person" /> <memberType
name="Group" /> -->
    <!-- Commented out - can't create in the sample -->
</createMemberType>
<updateMemberType>
    <!-- <memberType name="Person" /> <memberType name="Group" /
> -->
    <!-- Commented out - can't update in the sample -->
</updateMemberType>
<deleteMemberType>
    <!-- <memberType name="Person" /> <memberType name="Group"
/> -->
    <!-- Commented out - can't delete in the sample -->
</deleteMemberType>
<renameMemberType>
    <!-- <memberType name="Person" /> <memberType
name="Group" /> -->
    <!-- Commented out - can't rename in the sample -->
</renameMemberType>
<moveMemberType>

```

```

        <!-- <memberType name="Person" /> <memberType
name="Group" /> -->
        <!-- Commented out - can't move in the sample -->
</moveMemberType>

    <nodeMaps>
<nodeMap node="o=company,c=us" pluginNode="o=company,c=us" />
//Root DN
    </nodeMaps>
</profileRepository>

```

---

**Note:** Some WPSConfig steps may overwrite this file (the modified wmm.xml) so it is necessary to make a copy of this modified file. If you perform configuration steps, please check if the right file is in place.

---

14. In the file below, ensure that the following filters are correct for your environment and edit the values if needed.

For example:

```

WPS_install_dir\shared\app\config\services\PumaService.properties
user.base.attributes=cn,uid,cn,logonId,logonPasswordVerify,logonPassword ...
group.base.attributes=cn,uniqueOwnerIdentifier,membergroups,groupmembers,memberGroupName,memberGroupType,distinguishedName...

```

15. Ensure that the user.minimum.attributes values include all attributes for the user that the CMR will retrieve. For example:

```
user.minimum.attributes=cn,genUserid,cn,givenName,sn,mail...
```

16. Ensure that all user.minimum.attributes have the correct attribute access control for the Administrator and all are in one of the NetPoint panels, as described in the *NetPoint 7.0 Administration Guide Volume 1*.

For example:

```

COREid System Console > User Manager Configuration > Configure Tab >
Link > View Object Profile > Configure Panels

```

17. Configure WebSphere Portal security by executing the following command:

```
WPS_install_dir/config/WPSConfig.bat action-secure-portal-ldap
```

18. Make sure whether the correct wmm.xml file is in place.
19. Start WebSphere Portal server.

20. Make sure the `systemcred.dn` is valid in the following file:

`WPS_install_dir\shared\app\config\services\VaultService.properties` file

For example:

`systemcred.dn=cn=PortalAdmin,o=company,c=us`

---

**Note:** This is always the fully qualified `uniqueId` of `wpsadmin`.

---

21. Configure WebSphere Portal credentials by executing the following command  
`WPS_install_dir/config/WPSConfig.bat action-create-deployment-credentials`
22. Access `http://host:port/wps/portal`.  
where *host* is the fully qualified server name and *port* is the port number configured for the Portal Server.
23. Login to the Portal using Netpoint admin user. Login should be successful and Admin user should be able to search for other NetPoint Repository users and groups.

## Setting Up NetPoint with WebSphere Portal v5.1

Integrating the WebSphere Portal v5.1 for COREid involves a series of installation and configuration tasks.

### To integrate the WebSphere Portal with COREid

1. Complete the tasks in “Preparing to Install the NetPoint Connector” on page 190:
  - a) Install the WebSphere Application Server with appropriate Fix Pack as described in the WebSphere documentation to correct the following issues.

For example:

**Issue:** Access permission is not working for any user `dn` containing intermediate spaces after the comma separators, as in the following:  
`cn=PortalUser, o=company, c=us`.

Consult IBM Fix PQ93461 for details about this Portal User access permission problem.

**Action:** This IBM fix pack needs to be applied. Please verify the Portal Server History Log to ensure that all required fixes have been installed. Refer to the Portal Infocenter documentation for additional details.

- b) Install the WebSphere Portal with WebSphere Application Server security disabled.

---

**Note:** Both the Global and Java 2 security should be disabled while installing the Portal Server.

---

See the IBM WebSphere Portal Infocenter documentation for details on installation.

- c) Apply Fixes PQ99439 and PK02868\_510. This is required for custom user registry configuration of WebSphere Portal 5.1. (If these fixes are not applied, task enable-security-wmmur-custom will fail.)
  - d) Apply the latest available “Member Manager cumulative fix for WebSphere Portal version 5.1.” This includes a fix for the group membership feature. This cumulative fix is located at the following Web site:  
  
`http://www-1.ibm.com/support/docview.wss?rs=0&uid=swg24009153`
  - e) Install and configure NetPoint, as covered in “Preparing to Install the NetPoint Connector” on page 190.
2. Install the NetPoint Connector for WebSphere and configure the NetpointWASRegistry and TAI components, as described in “Installing the NetPoint Connector for WebSphere” on page 206.
  3. Complete NetPoint Connector setup and testing, as covered in “Completing NetPoint Connector Setup” on page 217.
  4. In the WebSphere Administration Console, change security to custom registry. This specifies the NetPointWASRegistry, which establishes a connection between the WAS and NetPoint. The WAS uses the NetPointWASRegistry to authenticate and authorize portal users through NetPoint security policies.

5. Ensure that the following Admin credentials are set in clear text in the `NetPointWASRegistry.properties` file:

```
OB_AdminUserName=admin  
OB_AdminUserCreds=password
```

where the `OB_AdminUserName` value is the userid of the Portal Server administrator who is a NetPoint Master Identity Administrator or NetPoint Administrator. This is required for the CMR. The Admin credentials should be set in clear text. The `NetPointWASRegistry` reads the password, encrypts it, and re-writes the properties file with the encrypted password. The encryptor can be executed by running the `registryTester` program, as well as from WebSphere. To assist you with adding these parameters, see the `NetPointWASRegistryProperties.sample` file, which includes comments. For additional details, consult, “`NetPointWASRegistry.properties`” on page 283.

---

**Note:** `NetPointWASRegistry.properties` file formatting is lost when NetPoint re-writes the file with the encrypted password. Therefore, you may want to save a copy of the `NetPointWASRegistry.properties`.

---

6. Restart the WAS Server.
7. Ensure that both the WebSphere Application server and Portal Server are stopped.
8. Make a back-up copy of the following file:

```
WPS_install_dir/config/wpconfig.properties
```

where `WPS_install_dir` is the directory where WebSphere Portal Server is installed.

**9. Edit the wpconfig.properties file and add the following:**

PortalAdminId (Example: PortalAdminId= DN of wpsadministrator)

PortalAdminIdShort (Example: PortalAdminId= wpsadministrator)

PortalAdminPwd (Example: PortalAdminPwd=wpsadminpassword)

WasUserId (Example: WasUserId=wasadministrator)

WasPassword (Example: WasPassword=wasadminpassword)

PortalAdminGroupId (Example: PortalAdminGroupId= DN of PortalAdminGroupId)

PortalAdminGroupIdShort (Example: PortalAdminGroupIdShort= PortalAdminGroupId)

LTPAPassword (Example: LTPAPassword= Password configured for LTPA in the AppServer Configuration)

WmmSystemId (Example: WmmSystemId= Set this value to same as that of PortalAdminIdShort)

WmmSystemIdPassword (Example: WmmSystemIdPassword= Set this value to same as that of PortalAdminPwd)

LDAPSuffix (Example: LDAPSuffix=o=company, c=us LDAP suffix of Netpoint installation)

LDAPUserSuffix (Example: LDAPUserSuffix=ou=users Keep this blank if user-nodes are directly under LDAPSuffix)

LDAPGroupSuffix (Example: LDAPGroupSuffix=ou=groups Keep this blank if group-nodes are directly under LDAPSuffix)

LdapUserPrefix (Example: LdapUserPrefix=cn)

LdapGroupPrefix (Example: LdapGroupPrefix=cn)

**10. Restart the WebSphere Application server.**

**11. Optional-Turn on PUMA traces in Portal, which can be done by entering the following in the log.properties file, which typical resides at**

WPS\_install\_dir\shared\app\config\log.properties

traceString=com.ibm.wps.services.puma.\*=all=enabled:com.ibm.wps.puma.\*=all=enabled:com.ibm.wps.command.puma.\*=all=enable

**12. Backup the following files:**

WPS\_install\_dir/wmm/wmm.xml

WPS\_install\_dir/wmm/wmm\_DB.xml

- 13. Make a copy of wmm.xml and save it under the name wmm\_custom.xml in the same folder as wmm.xml (WPS\_install\_dir/wmm/wmm\_custom.xml)**

**Edit wmm\_custom.xml, changing your Netpoint CMR configuration as follows. (Verify that the lookaside flag is set to "false").**

```
<supportedMemberTypes>

    <supportedMemberType name="Person"

        rdnAttrTypes="uid"

        defaultParentMember="o=company,c=us"

        defaultProfileRepository="CNR"/>

    <supportedMemberType name="Group"

        rdnAttrTypes="cn"

        defaultParentMember="o=company,c=us"

        defaultProfileRepository="CNR"/>

</supportedMemberTypes>

<profileRepository name="NetpointCustomRepository"

    UUID="CNR"

    description="This is Netpoint WMM custom MemberRepository
implementation." vendor="OBLIX!"

    adapterClassName="com.oblix.registry.NetPointMemberRepositoryImpl_
v51" //Name of the CMR implementation class

    specVersion="1.0" adapterVersion="2.0"

    configurationFile="customRepositoryAttributes.xml"

    wmmGenerateExtId="true" supportGetPersonByAccountName="false"

    supportDynamicAttributes="false"
profileRepositoryForGroups="CNR"

    enableTrace="true" PumaService.properties="WPSInstallDir/shared/
app/config/services/PumaService.properties" //Path of the PUMA
service file.
```

NetPointWASRegistry.properties="NetpointWASConnInstallDir\oblix\config\NetPointWASRegistry.properties"> //Path of the Netpoint WAS registry configuration file.

Unix-This path should be mentioned as "NetpointWASConnInstallDir/oblix/config/NetPointWASRegistry.properties":

```
<readMemberType>

<memberType name="Person" /> <memberTypename="Group" />

//Only read access to CMR

</readMemberType>

<createMemberType>

<!-- <memberType name="Person" /> <memberTypename="Group" /> -->

<!-- Commented out - can't create in the sample -->

</createMemberType>

<updateMemberType>

<!-- <memberType name="Person" /> <memberType name="Group" /> -->

<!-- Commented out - can't update in the sample -->

</updateMemberType>

<deleteMemberType>

<!-- <memberType name="Person" /> <memberType name="Group"/> -->

<!-- Commented out - can't delete in the sample -->

</deleteMemberType>

<renameMemberType>

<!-- <memberType name="Person" /> <memberTypename="Group" /> -->

<!-- Commented out - can't rename in the sample -->

</renameMemberType>

<moveMemberType>

<!-- <memberType name="Person" /> <memberTypename="Group" /> -->

<!-- Commented out - can't move in the sample -->
```

```

</moveMemberType>

<nodeMaps>

<nodeMap node="o=company,c=us" pluginNode="o=company,c=us" />
//Root DN

</nodeMaps>

</profileRepository>

```

- 14.** Overwrite following 2 files with new contents of wmm\_custom.xml.

```

WPS_install_dir/wmm/wmm.xml
WPS_install_dir/wmm/wmm_DB.xml

```

- 15.** Ensure that the following filters are correct for your environment in the following file:

```

WPS_install_dir\shared\app\config\services\PumaService.properties

```

Edit the values, as needed.

```

user.base.attributes=cn,uid,cn,logonId,logonPasswordVerify,logonPassword ...

group.base.attributes=cn,uniqueOwnerIdentifier,membergroups,groupmembers,memberGroupName,memberGroupType,distinguishedName ...

```

- 16.** Ensure that the `user.minimum.attributes` values include all attributes for the user that the CMR will retrieve.

**For example:**

```

user.minimum.attributes=cn,genUserId,cn,givenName,sn,mail...

```

- 17.** Ensure that all `user.minimum.attributes` have the correct attribute access control for the Administrator and all are in one of the NetPoint panels, as described in the NetPoint 7.0 Administration Guide Volume 1.

**For example:**

```

COREid System Console > User Manager Configuration > Configure Tab > Link > View Object Profile > Configure Panels

```

18. Modify the file `WPS_install_dir/wmm/wmmWASAdmin.xml` as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<wmmWASAdmins>
```

```
  <admin logonId="PortalAdmin" logonPassword="oblixoblix"
  uniqueUserId="cn=Portal Admin,o=company,c=us"/>
```

```
</wmmWASAdmins>
```

`logonId`, `logonPassword` and `uniqueUserId` correspond to the new values of `PortalAdminIdShort`, `PortalAdminId`, and `PortalAdminPwd` respectively in `wpconfig.properties`.

19. Modify `WPS_install_dir/wmm/wmmur.xml`. Set the value of `wmmnode` to your `LDAPSuffix` as follows:

```
<node wmmnode="o=company,c=us"/>
```

20. Modify `WPS_install_dir/wmm/wmmAttributes.xml` so that it contains only supported attributes. `wmmAttributes.xml` contains definitions of all the supported attributes. Each attribute definition might have two properties named `applicableMemberTypes` and `requiredMemberTypes`. Values of these properties should be supported member types. For example, if `wmm_custom.xml` has only two supported member types (`Person` and `Group`), then other types such as `organizationalUnit` or `organization` should not be `applicableMemberTypes` or `requiredMemberTypes` for any of the attributes in `wmmAttributes.xml`.

21. Modify `WPS_install_dir/shared/app/config/services/VaultService.properties`.

Set the value of `systemcred.dn` to the DN of `wpsadministrator`.

**Example:**

```
systemcred.dn=cn=Portal Admin,o=company,c=us
```

22. Configure WebSphere Portal security by executing the following command:

```
WPS_install_dir/config/WPSConfig.bat
enable-security-wmmur-custom
```

23. Verify that the correct `wmm.xml` file is in place. The contents of `wmm.xml` should match the contents of `wmm_custom.xml`.

24. Ensure that `systemcred.dn` is valid in the following file:

```
WPS_install_dir\shared\app\config\services\VaultService.properties file
```

**Example:**

```
systemcred.dn=cn=Portal Admin,o=company,c=us
```

---

**Note:** This is always the fully qualified `uniqueId` of `wpsadmin`.

---

25. Access the following Web page:

```
http://host:port/wps/portal
```

where host is the fully qualified server name and port is the port number configured for the Portal Server.

26. Login to the Portal as the Netpoint admin user. The Login should succeed, and Admin user should be able to search for other NetPoint Repository users and groups.

## Managing Users and Groups with Portal v5

Portal Administrators can use the NetPoint COREid System to perform user and group management tasks such as adding or deleting users and groups, modifying user profiles and attributes.

To use NetPoint's user and group management functionality, ensure that you do not create users and groups in the WebSphere Portal. Instead, create and modify users and groups in the NetPoint COREid System.

You can add and delete static groups and user membership in groups through the NetPoint COREid System. The information that you update in NetPoint is immediately reflected in the WebSphere Portal.

---

**Note:** To recognize group membership, NetPoint requires the dynamic group to be expanded.

---

After you create users and groups in the COREid System, you can search for them in the WebSphere Portal.

See the *NetPoint 7.0 Administration Guide Volume 1* for more information on managing users and groups.

## **Modifying User Profiles and Attributes**

When users modify their profile through the NetPoint COREid System, the modifications are immediately visible in the WebSphere Portal. This ensures that the most current user information is used when portal developers personalize user pages.

You can map additional attributes to a use's profile if necessary. See the WebSphere Portal documentation for information on mapping attributes.

## **Password Management with Portal v5**

Because the portal uses NetPoint SSO, users are subject to the NetPoint password policies during authentication.

---

**Important:** To implement the NetPoint password management feature, turn off the portal's password management functionality.

---

The NetPoint password management functionality includes defining password policies, resetting passwords, expiration notification, and challenge phrases for lost passwords.

See the *NetPoint 7.0 Administration Guide Volume 1* for more information on password policies.

## **Access Control for the WebSphere Portal v5**

Portal administrators use the portal's access control functionality to grant access to portlets. From the WebSphere Portal, administrators can search for

NetPoint-managed users and groups to whom they want to grant portal administration privileges as well as portlet access control.

## Configuring Single Sign-on Functions for the Portal v5

Configuring SSO between NetPoint and the WebSphere portal enables the WebSphere portal to utilize the ObSSOCookie and enable NetPoint Connector for WebSphere to authenticate NetPoint users.

Configuring SSO logout for the WebSphere Portal Server ensures that when a user logs out of a NetPoint-protected WebSphere resource, both the LTPA token and the ObSSOCookie are killed. The user will not be able to access any other WebSphere resource or other NetPoint-protected resources without authenticating again.

- If you have configured the TAI for single sign-on between NetPoint and WebSphere, you must configure single sign-on logout for the WebSphere Portal Server.
- If you have not configured the TAI for single sign-on, users can use the portal's logout button to log out of all NetPoint-protected resources.

### To configure SSO for the WebSphere Portal v5

1. Install the WebGate plug-in for the Web server that you selected when you installed the WebSphere Portal.
2. In the NetPoint Access Manager, define the URL that you want to protect.

A WebGate prompts for authentication when users attempt to log in to this URL. Be sure to protect / if you want the WebGate to prompt for authentication when the user gets to the root of the WebSphere Portal. You can also add other authorization rules, if needed.

---

**Note:** To protect resources, Oblix recommends that you use a form-based authentication scheme. However, if you use the basic authentication scheme, set the Challenge Redirect field to another WebGate to ensure that the ObSSOCookie gets set. See the *NetPoint 7.0 Administration Guide Volume 2* for more information on authentication schemes.

---

### To configure SSO logout

1. Create a NetPoint policy with a Form Over LDAP type of authentication scheme to protect the portal URL, as described in the *NetPoint 7.0 Administration Guide Volume 2*.
2. Create a custom logout page using HTML, JSP, or CGI protocol.

The default logout page for NetPoint, `logout.html`, is located in:

`WebGate_install_dir\access\oblix\apps\common\bin`

where `WebGate_install_dir` is the directory where the WebGate is installed.

3. Save the logout page in the document root of the Web server on which the WebGate that protects WebSphere is installed.

For example:

`http://foobar/myportal/logout.html`

---

**Note:** Ensure that the name of the logout page contains the string “logout”.

---

4. Protect the logout page with a NetPoint policy, as described in the *NetPoint 7.0 Administration Guide Volume 2*.
5. Locate and open the file below:  
`WPS_install_dir\shared\app\config\services\ConfigServices.properties`  
where `WPS_install_dir` is the directory where WebSphere Portal Server is installed.
6. Add the following two parameters in `ConfigServices.properties` file:  
`redirect.logout = true`  
`redirect.logout.url = The path to the logout page.`  
For example:  
`http://foobar/myportal/logout.html`
7. Restart the WebSphere Portal Server and Application Server.

## Configuring the WebSphere Application Server v6

The following sections describe how you integrate COREid with the WebSphere Application Server v6:

- “Enabling the NetPointWASRegistry for WAS v6” on page 277
- “Testing the NetPointWASRegistry for WebSphere v6” on page 278
- “Configuring the TAI for WebSphere v6” on page 279
- “Testing the TAI for WAS v6” on page 282
- “Enabling Logging for TAI for WAS v6” on page 283

Before you begin, see Table 10, “COREid Versions Supported for Specific WebSphere Products on Specific Operating System Platforms,” on page 189 for complete details about version support.

## Enabling the NetPointWASRegistry for WAS v6

After you have installed WAS v6 and the NetPoint Connector for Websphere, then tested them to be sure they are communicating, you can enable the NetPointWASRegistry, which interoperates with the User Registry in WebSphere 6. This allows NetPoint to act as the authentication source for the WebSphereApplication Server.

### To enable the NetPointWASRegistry in WAS 6

1. Ensure that WebGate protecting the WebPass has IPValidation set to False.
2. Make a backup copy of the `security.xml` file located in the following directory:

```
WAS_install_dir/profiles/default/config/cells/serverNodeName
```

---

**Note:** Create a backup copy of `security.xml` whenever you make a change to the configuration. If you make errors in the new configuration, you can always restore the previous version of the `security.xml` file.

---

3. Start the WebSphere Admin Service.
4. Log into the WebSphere Administration Console as the Identity Administrator.

---

**Note:** Any NetPoint user added to the role "Admin" must belong to the group called "Admin" or "Administrator." Otherwise, this user may not be able to login to WebSphere Administrative Console. Other roles, such as "monitor" have no restrictions. Thus, any NetPoint user added to these roles through the WAS 6 Administrative Console can log into the WAS 6 Administrative Console.

---

The Administrative Console for WAS 6 is accessible through the following URL:

```
http://hostname.domain.com:9060/ibm/console
```

where `hostname` is the fully qualified name of the machine where WebSphere is installed; for example, `xyz.domain.com`.

5. Navigate to Security > Global Security > Custom and enter values for the following variables:
  - **Server user ID**
  - **Server User Password**
  - **CustomRegistry Classname**
  - **NetPoint Admin ID**
  - **User password**
  - **NetPoint Classname:** `com.oblix.registry.NetPointWASRegistry`

6. Navigate to Configuration > Additional Properties > Custom Properties > New, enter values for the following parameters, then click Apply to save your changes.
  - **Name:** NetPointWASRegistry.properties
  - **Value:** /opt/netpoint/oblix/config/NetPointWASRegistry.properties
  - **Description:** Property file for NetPoint User Registry
7. Navigate to Global Security > Authentication Mechanisms > LTPA.
8. In the Configuration tab, specify a password, then click OK.
9. Click Single Signon (SSO).
  - Click the Enabled box.
  - Enter the domain name (for example, oracle.com)., then click Apply.
10. In the navigation pane on the left, click Security > Global Security, then complete the following steps:
  - a) Set the Active Authentication Mechanism to LTPA.
  - b) Set the Active User Registry to Custom User Registry, then click OK.
  - c) Click the Enable Global Security box.
  - d) Click Apply to test the configuration. If the information is valid, a message confirming the changes displays at the top. Otherwise, correct any errors that are reported.
  - e) Click Save.
  - f) Click Logout and close the browser window.
  - g) Stop the WebSphere Application Service. If a message announces that Service could not be stopped, switch to the Task Manager and ensure that no Java processes are currently running.
11. Start the WebSphere Administration Service.

## Testing the NetPointWASRegistry for WebSphere v6

After you have enabled the NetPointWASRegistry, you need to verify that it is configured correctly.

## To test the NetPointWASRegistry configuration

1. Access the Snoop Servlet sample running on the default server at the following URL:

```
http://hostname:9080/snoop
```

where *hostname* is the fully qualified name of the machine where WebSphere is installed.

Alternatively, you can use the following URL:

```
http://hostname:web_server_plug-in_port/snoop
```

where *hostname* is the fully qualified domain name of the machine on which WebServer is installed.

2. When challenged by WebSphere, enter a username and password that is valid in NetPoint.

By default, any authenticated user should be allowed access.

3. Launch the WebSphere Administrative Console and login as the user specified in Security > Global security > Custom.

If the configuration is valid, the login will succeed.

4. Set access control for the WebSphere Administrative Console by specifying the users and groups and the roles to which they belong.

See your WebSphere 6 documentation for details.

5. Write down the names of the .xml files that you have modified to support Admin Console access, then click Save.

After you have installed NetPoint Connector for WebSphere, you must configure the NetPointWASRegistry and TAI. The NetPointWASRegistry handles authentication, and the TAI facilitates NetPoint single sign-on.

## Configuring the TAI for WebSphere v6

You configure the TAI to enable single sign-on between NetPoint and WAS. For WebSphere 6.0 you must install *webgate.properties*, add the TAI, and then add *custom.properties*.

## To install and configure TAI for WAS 6

1. Copy the configuration file `webgate.properties` from the following directory :

```
NPCWS_install_dir/oblix/config
```

to the WebSphere installation properties directory, which resides in the following location:

```
WAS_install_dir/properties
```

where `NPCWS_install_dir` is the directory where NetPoint Connector for WebSphere is installed, and `WAS_install_dir` is the directory where the WebSphere Application Server is installed.

`webgate.properties` contains configuration information that WebSphere will use to connect to the AccessGate.

2. In the WebSphere installation properties directory, modify the parameter values in the `webgate.properties` file ) as follows:

```
OB_InstallDir = NPCWS_install_dir
```

where `NPCWS_install_dir` is the directory where NetPoint Connector for WebSphere is installed.

If the associated WebGate is installed on a proxy server used as a front end server to direct all user requests to Web servers that interface with WebSphere Application Servers, then specify the following parameter values:

- `OB_IsProxyEnabled=true`
- `OB_hostnames = serverName`

where `serverName` is the name of the proxy server.

- `OB_ports = portNumber`

where `portNumber` is the port number of the proxy server.

---

**Note:** If you used a resource other than Authen, you must specify the `resourcename` in the `webgate.properties` file.

---

3. Launch the WebSphere Administrative Console
4. In the navigation pane on the left, navigate to Global security > Select Authentication Mechanisms > LTPA > Trust Association > Additional Properties > Interceptors > New.
5. In the Name field, enter `Oblix TAI Interceptor`.
6. In the Interceptor classname field, enter `com.oblix.tai.was5.WebGateTrustAssociationInterceptor`, then click OK.

**7. Make the following changes:**

a) **Name:** `com.ibm.websphere.security.trustassociation.types`

- **Value:** `webgate`
- **Description:** NetPoint TAI property

Select the Required checkbox, then click OK.

b) **Name:** `com.ibm.websphere.security.trustassociation.webgate.config`

- **Value:** `webgate`
- **Description:** Name of the NetPoint TAI properties file located in *WAS\_install\_dir/properties* directory.

Select the Required checkbox, then click OK.

c) **Name:** `com.ibm.websphere.security.trustassociation.webgate.interceptor`

- **Value:** `com.oblix.tai.was5.WebGateTrustAssociationInterceptor`
- **Description:** NetPoint TAI class for WebSphere 6

Select the Required checkbox, then click OK.

**8. Click Interceptors at the top of the page and then click Save.**

**9. Navigate to the following:**

`WAS_install_dir/profiles/default/config/cells/serverNodeName`

**10. Make a backup copy of the `security.xml` file.**

**11. In the WebSphere Administrative Console, navigate to LTPA > Trust Association > Interceptors.**

**12. Select the WebSeal Interceptor, then click Delete.**

**13. Click Trust Association, then click the Enable Trust Association check box.**

**14. Click Apply, then click Save.**

**15. Logout of the WebSphere Administrative Console, then close the browser.**

**16. Shut down the Websphere Admin Server.**

If you get an error message, go to Task Manager and ensure that no Java process is running.

**17. Restart the WebSphere Admin Service.**

If the Service does not start, verify that the properties are set correctly in the `security.xml` file and that the `webgate.properties` file is in the correct location.

18. Create a NetPoint policy to protect WebSphere resources as described in “To install and configure TAI for WAS 4” on page 227.

To facilitate access the Administration Console after you have enabled TAI, you need to enable the Policy created in “Defining a Policy Domain for WebSphere” on page 201. You also need to enable the policy created in “Defining a Policy Domain for the WebSphere v6.0 Administration Console” on page 204.

22. Verify that the TAI is working, as detailed below.

## Testing the TAI for WAS v6

### To test the TAI

After you have configured TAI, test for successful authentication and single sign-on between WebSphere and NetPoint.

To conduct these tests, use the Snoop servlet that WebSphere provides. The Snoop servlet has security constraints that only allow access to authenticated users.

When WebSphere security is not enabled, access to the Snoop is unrestricted.

When WebSphere security and TAI are enabled, users attempting to access Snoop will be challenged by NetPoint. If TAI is not enabled, users attempting to access Snoop will be challenged by WebSphere as well.

### To test NetPoint single sign-on for NetPoint-protected WebSphere resources

1. On the Web server you use to access the WAS, navigate to the document root and create a directory named test.
2. In the test directory, create a file named index.html.
3. In NetPoint, create and enable policies to protect the `/snoop` and `/test` directories.
4. Access the Snoop servlet through the following URL:

`http://hostname.domain.com:80/snoop`

where *hostname* is the fully qualified name of the machine where WebServer is installed.

You will be challenged for authentication. After you are authenticated, you will be allowed to access the Snoop servlet.

5. Access the `/test` URL.

Verify that you can access the URL and view the index.html file without being challenged.

## Enabling Logging for TAI for WAS v6

You can enable logging for TAI from the WebSphere Administrative Console.

### To enable logging for TAI for WAS 6

1. Launch the WebSphere Administrative Console.
2. Navigate to Troubleshooting > Logs and Trace
3. Select your Server.
4. Select Change Log Level Details.
5. Select Components.
  - Enable debug logging for  
`com.oblix.tai.was5.WebGateTrustAssociationInterceptor`

## Configuration Files

The following configuration files are used when integrating NetPoint Connector for WebSphere with WAS:

- NetPointWASRegistry.properties
- webgate.properties
- TrustedServers.properties

### NetPointWASRegistry.properties

Table 11 describes the parameters of NetPointWASRegistry.properties file located in *NPCWS\_install\_dir/oblix/config*. This file contains data that was specified during NetPointWASRegistry installation, as well as some default parameter values for logging. For example:

```
# Logging level (none, info or debug);  
OB_LogLevel=debug  
OB_LogFileName=C:/NPCWS_install_dir/log
```

---

**Note:** Webpass to GroupSrvCenter performance has been improved with the addition of configuration options to improve IdentityXML calls in NetPoint. For example, when no nested groups are used and are turned off, you may use a new option so that getGroups will not generate a request for nested groups.

---

See also the NetPointWASRegistryProperties.sample file.

**Table 11** Parameters in NetPointWASRegistry.properties

Parameter Name	Description	Optional/ Mandatory
<b>Installation</b>		
OB_InstallDir	The directory where NetPointWASRegistry is installed.	Mandatory
<b>Logging</b>		
OB_LogLevel	The logging level that is recorded in the log file. Values are none, info, and debug.	Optional
OB_LogFileName	The file name for Custom User Registry (NetPointWASRegistry) log messages. Default = <i>NPCWS_install_dir/log</i> <b>Note:</b> Log messages for the CMR are directed to the <i>WPS_install_dir/log/appserver-out.log</i> file.	Optional
OB_LogMilliseconds=true	The data/time format of log messages in the file specified with OB_LogFileName. When true, log messages are time formatted in milliseconds. Default =true	Optional
<b>WebPass</b>		
OB_WebPassHost	The WebPass server host machine name. The host name must be fully qualified; for example, OB_WebPassHost=hostname.acme.com.  To configure multiple WebPass instances for failover purposes, separate the names with a comma. For example:  OB_WebPassHost=foo.domain.com , bar.domain.com  Note that the host name corresponds to the port number in the specified order. See the example in the Ob_WebPassPort description section below.	Mandatory

**Table 11** Parameters in NetPointWASRegistry.properties

Parameter Name	Description	Optional/ Mandatory
OB_WebPassPort	<p>The port number of the host machine.</p> <p>To configure multiple WebPass instances, separate the port numbers with a comma. For example:</p> <p>OB_WebPassPort=80, 81</p> <p>Note that the host name corresponds to the port number in the specified order. In the above example, the hostname:port number pairing is as follows:</p> <p>foo.domain.com:80 bar.domain.com:81</p> <p>For failover to work, all other variables such as user name, credentials and webgate protection must be the same.</p>	Mandatory
OB_WebPassIsProtected	<p>Values are true and false. If WebPass is protected, set value=true.</p>	Mandatory
OB_AdminUserName	<p>NetPoint requires the Admin username and password to make IdentityXML calls to the WebPass. For details about administrator rights, see “Configuring the COREid Server for WAS Search Methods and the NetPointWASRegistry Admin” on page 155.</p>	Mandatory
OB_AdminUserCreds	<p>NetPoint requires the Admin username and password to make IdentityXML calls to the WebPass. Without the password the connector will not work.</p> <p><b>Note:</b> You need to enter a clear-text password, which the program will encrypt and re-write to the properties file after the first run.</p>	Mandatory

**Cookie**

**Table 11** Parameters in NetPointWASRegistry.properties

Parameter Name	Description	Optional/ Mandatory
OB_CookieDomain	The cookie domain specified in the NetPoint WebGate installer configuration. Needed if WebPass is protected.  For example, .xyz.com	Mandatory
OB_CookiePath	The cookie path specified in the NetPoint WebGate configuration. Needed if WebPass is protected. Default = /	Mandatory
<b>WebPass SSL</b>		
OB_WebPassSSLEnabled	Specifies whether WebPass needs HTTPS connection. Values are true and false. Default = false	Mandatory
<b>Login and Search Attributes</b>		
OB_UserAttr	The unique user identification (for example, uid)	Mandatory
OB_UserSearchAttr	The DN prefix for users from LDAP (for example, cn)	Mandatory
OB_GroupSearchAttr	The DN prefix for groups from LDAP (for example, cn)	Mandatory
<b>Active Directory Forest</b>		
OB_WebPassADDomain	Optional. The domain of the Admin user. To be used in case of Active Directory Forest with multiple domains.  For example, OB_WebPassADDomain=ou=company,dc=qalab,dc=acme,dc=com  The ADDomain must be the same as the default defined in the COREid Server.	Optional

**Table 11** Parameters in NetPointWASRegistry.properties

Parameter Name	Description	Optional/ Mandatory
<b>Records Returned</b>		
OB_WebPassXPIRecordsReturned	Optional. The number of records to return for getUsers or getGroups. This is used only in the WebSphere Portal Default = return all	Optional
<b>Authentication</b>		
OB_AuthnSchemeResourceTypeName	Authen	Mandatory
OB_AuthnSchemeOperation	LOGIN	Mandatory
OB_AuthnSchemeResourceName	/Authen/Basic	Mandatory
OB_AuthzActionType	WAS_Registry	Mandatory
OB_AuthzActionName	uid	Mandatory
<b>Cache</b>		
OB_AllUserCache_enabled	Enables caching of all users. Values are true and false.	Optional
OB_AllUserCache_timeout	Timeout for cache of list of all users.	Optional
OB_UserAttributesCache_enabled	Enables Caching of UserAttributes. Values are true and false.	Optional
OB_UserAttributesCache_timeout	The timeout for the cache of user attributes. Timeout is for the whole cache.	Optional
OB_UserAttributesCacheElement_timeout	The timeout for the cached user attributes. The Timeout is per user.	Optional
OB_GroupAttributesCache_enabled	Enables Caching of GroupAttributes. Values are true and false.	Optional

**Table 11** Parameters in NetPointWASRegistry.properties

Parameter Name	Description	Optional/ Mandatory
OB_GroupAttributesCache_timeout	The timeout for the cache of group attributes. Timeout is for the whole cache.	Optional
OB_GroupAttributesCacheElement_timeout	The timeout for the cached group attributes. The Timeout is per group.	Optional
OB_AllGroupCache_enabled	Enables caching of list of all groups. Values are true and false. Used only for all groups, and mostly used by the Admin Console.	Optional
OB_AllGroupCache_timeout	The timeout for cache of the list of all groups.	Optional
OB_UserGroupsCache_enabled	Enables caching of list groups of which the user is a member. Values are true and false. Maintains a cache of all the groups a logged in user belongs to.	Optional
OB_UserGroupsCache_timeout	The timeout for cache of the list of groups for a user. The timeout is per user. This value should not be very high--if the user's group membership changes the new membership will only take affect at cache timeout. For example, a value of 3600 equates to 1 hour.	Optional
OB_GroupMembersCache_enabled	Enables caching of list of groups and list of members in each group. Values are true and false. Stores members for each groups (not a frequently used cache).	Optional
OB_GroupMembersCache_timeout	Specifies the timeout for cache of list of groups and the list of members in each group.	Optional

**Keystore**

**Table 11** Parameters in NetPointWASRegistry.properties

Parameter Name	Description	Optional/ Mandatory
OB_Keystore	<p>Specifies the keystore file used by the registry when it makes SSL connections to HTTPS WebPass. The keystore contains the requestor's public and private key pairs, X.509 certificate, and certificates for Certificate Authorities trusted to certify responder servers. The keystore is managed using the JDK keytool.</p> <p>For example:  <i>NPCWS_install_dir/oblix/config/jssecacerts</i></p>	Optional
OB_KeystorePassword	Optional. The password for the keystore.	Optional
<b>Users and Groups</b>		
OB_UserTabId	<p>For future use. Do not change the default.                      Default = Employees</p>	Mandatory
OB_GroupTabId	<p>For future use. Do not change the default.                      Default = Groups</p>	Mandatory
<b>Performance</b>		
OB_NestedGroupsEnabled	<p>Values are true and false. The default is true.</p> <p>To improve GroupSrvCenter performance when nested groups are <i>not</i> used, set the value to false.</p> <ul style="list-style-type: none"> <li>• Nested groups will not be included in the search; the uniquemember attribute will not be requested in a group search when OB_NestedGroupsEnabled=false.</li> <li>• A value of true retrieves the uniquemember attribute in the group search, uses this for nested group computation, then removes it before the group is recorded.</li> </ul>	Optional

**Table 11** Parameters in NetPointWASRegistry.properties

Parameter Name	Description	Optional/ Mandatory
OB_DynamicGroupsEnabled	Values are true and false. To improve GroupSrvCenter performance when you are not using dynamic groups, set the value to false. Dynamic groups will not be included in the search.	Optional
<b>Non-Unique Login ID in Different Domains</b>		
OB_DnlsUniquelIdentifier= <b>See also</b> OB_DnlsUniquelIdentifier= in "WebGate.properties Configuration File" on page 290	Values are true and false. The default is false. Set the value to true to enable the NetPoint Connector for WebSphere to work in a multi-domain directory server with a non-unique login ID in the different domains.	Optional
OB_WebGateTAIEnabledForWAS4=true <b>See also</b> OB_DnlsUniquelIdentifier= in "WebGate.properties Configuration File" on page 290.	Values are true and false. The default is true and assumes the WAS 4 TAI is being used. Used in combination with OB_DnlsUniquelIdentifier=true to enable the NetPoint Connector for WebSphere to work in a multi-domain directory server with a non-unique login ID in the different domains.	Optional

## WebGate.properties Configuration File

Table 12 describes the parameters of the webgate.properties file. This file is located in *NPCWS\_install\_dir/oblix/config* with a copy in *WAS\_install\_dir\properties*. See also the webgateProperties.sample file.

**Table 12** Parameters in webgate.properties

Parameter	Description
OB_InstallDir	The directory where the NetPointWASRegistry is installed. Default = <i>NPCWS_install_dir</i>
OB_ISPROXYENABLED	Not required unless you use a proxy server. The default value is false. If you use a proxy server the value must be changed to true.

**Table 12** Parameters in webgate.properties

OB_hostnames	Not required unless you use a proxy server. The name of the host machine. This is only used for proxy servers.
OB_ports	Not required. The port number of the host machine.
Ob_loginID	Not required.
OB_AuthnSchemeResourceTypeName	Authen
OB_AuthnSchemeOperation	LOGIN
OB_AuthnSchemeResourceName	/Authen/Basic
OB_AuthzActionType	WAS_Registry
OB_AuthzActionName	uid
OB_DnIsUniquelIdentifier	Not required unless you want to enable the NetPoint Connector for WebSphere to work in a multi-domain directory server with a non-unique login ID in the different domains. See also, Non-Unique Login ID in Different Domains in "NetPointWASRegistry.properties" on page 283.

## TrustedServers Configuration File

Table 13 describes the parameters of the trustedservers.properties configuration file.

**Table 13** Parameters in TrustedServers.properties

Parameter	Description
com.ibm.websphere.security.trustassociation.enabled	true
com.ibm.websphere.security.trustassociation.types	webgate
com.ibm.websphere.security.trustassociation.webgate.interceptor	com.oblix.tai.WebGateTrustAssociationInterceptor
com.ibm.websphere.security.trustassociation.webgate.config	webgate

## Implementation Notes for the TAI

The following implementation is *optional* to enable the NetPoint Connector for WebSphere to work in a multi-domain directory server with a non-unique login ID in the different domains.

To accomplish this optional implementation, you use two parameters in the NetPointWASRegistry.properties file:

```
OB_DnIsUniqueIdentifier=
```

The default is false. Be sure to set OB\_DnIsUniqueIdentifier to true if the DN is being used.

In addition, you need to use this parameter in the NetPointWASRegistry.properties file:

```
OB_WebGateTAIEnabledForWAS4=true
```

The default assumes the WAS 4 TAI is being used. Be sure to set OB\_WebGateTAIEnabledForWAS4 to false if WAS 5 TAI is being used.

For more information, see “NetPointWASRegistry.properties” on page 283.

You also need the parameter below, in the webgate.properties file, to enable the NetPoint Connector for WebSphere to work in a multi-domain directory server with a non-unique login ID in the different domains:

OB\_DnIsUniqueIdentifier=false

This optional parameter is used when the TAI module is configured to pass on the users DN instead of the userAttr or LoginID. The default is false. If the OB\_DnIsUniqueIdentifier parameter is set to true, the DN is used to communicate between the TAI and Registry. Be sure to set the OB\_DnIsUniqueIdentifier to true if the DN is being used.

---

**Note:** The NetPointRegistry.properties and webgate.properties files must be synchronized.

---

The optional implementation described above works with the following caveats:

- Using TAI or WebGate as the only means of authentication should *not* be an issue since it is likely to be a requirement for multi-domain authentication. No users can go directly to WAS applications.

---

**Note:** An exception is when logging into the NetPoint Administration Console.

---

- A unique ID is used for the WAS\_ADMIN Account across all domains.
- This solution causes a loss of functionality regarding mapping individual users to roles.

**On WAS 4**—Role mapping can be done directly for users.

**On WAS 5**—Role mapping can be done only through NetPoint groups. This restriction is only present for WAS5.

# Implementation Notes for Active Directory

The following are issues to consider when implementing the NetPoint Connector for WebSphere on Active Directory.

## Configuring NetPoint Connector for WebSphere for Active Directory Forest

The steps to configure NetPoint Connector for WebSphere for an Active Directory Forest follow.

### To configure the NetPoint Connector for an Active Directory forest

1. In the NetPoint Access System Console, create a new Basic Over LDAP authentication scheme for a domain in the Active Directory Forest.

The base credentials that you specify in the Plugin(s) field must be the same as the search base that you specified in the directory server profile.

Details for Authentication Scheme			
<b>Name</b>	WebSphere Basic over LDAP		
<b>Description</b>	This scheme is basic over ldap		
<b>Level</b>	1		
<b>Challenge Method</b>	Basic		
<b>Challenge Parameter</b>	realm:NetPoint Basic Over LDAP		
<b>SSL Required</b>	No		
<b>Challenge Redirect</b>			
<b>Plugin(s)</b>	<b>Order</b>	<b>Plugin Name</b>	<b>Plugin Parameters</b>
	1	credential_mapping	obMappingBase="ou=company,dc=rhodium,dc=acme,dc=com",obMappingFilter="(objectclass=User)(samaccountname=%userid%)"
	2	validate_password	obCredentialPassword="password"
<input checked="" type="checkbox"/> Update Cache			

If you already created an administrator during pre-installation setup, you do not need to complete step 2. See “Preparing to Install the NetPoint Connector” on page 190 for more information.

2. Create a WebSphere administrator in NetPoint with View and Delegated Administration rights.

Ensure that the administrator’s login identification is unique.

3. Specify the WebSphere administrator as the administrator for the Active Directory forest domain.

This domain must be the same as the one for which you created the authentication scheme in Step 1. To do this, specify values for the `OB_WebPassADDomain` parameter in the `NetPointWASRegistry.properties` file as described in Table 11 on page 284.

You can search for users in the parent domain but you cannot search for users in sibling or children domains.

---

**Note:** You do not need to create an administrator for every domain in an Active Directory Forest.

---

## Set Active Directory Domain in `NetPointWASRegistry.properties`

If you are running Active Directory using multiple domains, you must manually edit the `NetPointWASRegistry.properties` file to include a value for the `OBWebPassADDomain` parameter.

For example, `OBWebPassADDomain=dc=xyz, dc=acme, dc=com`

The domain must be the same as the domain defined for the default directory server in the COREid System.

See the *NetPoint 7.0 Administration Guide Volume 1* for more information.

## Troubleshooting the NetPoint Connector for WebSphere

The following is a list of the most frequently asked questions on NetPoint Connector for WebSphere. See also, “Troubleshooting the NetPoint Connector for Portal Server v5” on page 306.

1. **Problem**—I am locked out of WAS 5 Admin Console or the WebSphere 5 Server does not start after making a configuration change. What should I do?

**Solution**—Restore the previous WAS 5.0 `security.xml` file located in `WAS_install_dir/config/cells/serverName` directory. This assumes you have made a backup of an older, working copy.

2. **Problem**—On Solaris, when setting up the SSL connector for NetPoint Connector for WebSphere, why does the keytool command give a “Signature not available” error?

**Solution**—This is a jdk 1.2.x problem. Use the NT version or any other jdk1.3.x version to create the cert db (jssecacerts) and then use it with WebSphere on Solaris.

3. **Problem**—Why do I get “All the jars are not in classpath: NoClassDefException”?

**Solution**—Make sure that the NetPointWASRegistry.jar and jobaccess.jar are in the classpath.

4. **Problem**—Why do I get SSLPeerUnverifiedException - peer not authenticated exception?

**Solution**—The jvm being used is different from the jvm that has imported the certificates of ca and server. The jvm and keytools used must be from the same installation. If one keytool is used to add certificates and java is invoked from the other installation directory, then the jvm will not be able to use the certificates and will produce this exception.

5. **Problem**—Why do I get “ObConfig.NO\_CONFIG\_FILE”?

**Solution**—This error means that the Access SDK client configuration file is not found. Check the Install\_Dir parameter in the NetPointWASRegistry.properties file and ensure that the following points to the NetPointWASRegistry installation directory. For example:

```
# Installation directory of NetPointWASRegistry
OB_InstallDir=/NPCWS_install_dir/oblix/config/
```

6. **Problem**—Why do I get UnsatisfiedLinkError?

**Solution**—You probably do not have the Access SDK lib in the PATH or LD\_LIBRARY\_PATH depending on the platform.

For example:

- **On NT**—set the PATH as follows:

```
set PATH=%PATH%;c:\NPCWS_install_dir\oblix\lib
```

- **On Solaris**—set LD\_LIBRARY\_PATH as follows:

```
setenv LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/ NPCWS_install_dir/
oblix/lib export LD_LIBRARY_PATH
```

This can be either done at system level or at start-up.

- **For AIX**—set LIBPATH as follows:



**11. Problem**—Why do IdentityXML calls fail with an unauthorized exception?

**Solution**—Check the NetPointWASRegistry.properties file to ensure that the WebPass host name is fully qualified.

For example:

```
OB_WebPassHost=foobar.oblix.com
```

```
OB_WebPassPort=80
```

**12. Problem**—Why do I get a "server specific error 10" while restarting the WebSphere Administrative Server.

**Solution**—Ensure all Java process are killed before re-starting the WebSphere Administration Server.

**13. Problem**—I see the following exception:

```
Exception in thread "main" com.ibm.websphere.security.  
CustomRegistryException: admin at  
com.oblix.registry.NetPointWASRegistry.getUserDisplayName(NetPo  
intWASRegistry.java:622) at  
com.oblix.tools.registryTester.main(registryTester.java:69)
```

**Solution**—There can be many reasons for this exception. In the NetPointWASRegistry file, turn on the debug flag and check the NetPoint debug log path as shown below.

```
OB_LogLevel=debug
```

```
OB_LogFileName=/oblix/NetPointWASRegistry/log
```

**14. Problem**—I get the following error in the NetPoint log file:

```
Mon Jan 06 14:57:21 PST 2003: Error making SOAP request  
java.io.FileNotFoundException: http://cobalt.oblix.net:80/  
identity/oblix/apps/userservcenter/bin/userservcenter.cgi at  
sun.net.www.protocol.http.  
URLConnection.getInputStream(URLConnection.java:529) at  
com.oblix.soapclient.OblisSoapClient.doRequest(OblisSoapClient.  
java, Compiled Code)  
com.oblix.registry.NetPointWASRegistry.realGetUserDisplayName(N  
etPointWASRegistry.java:650)at  
com.oblix.registry.NetPointWASRegistry.getUserDisplayName(NetPo  
intWASRegistry.java:607) at  
com.oblix.tools.registryTester.main(registryTester.java,  
Compiled Code).
```

**Solution**—Make sure that the WebGateStatic.lst file, which is in \$WebGate\_install\_dir/access/oblix/apps/webgate directory, has the following line:

```
IPValidation:false
```

**15. Problem**—I get the following error in the NetPoint log file:

```
Mon Jan 20 15:37:24 GMT-06:00 2003: Error making SOAP request
java.io.IOException: Server returned HTTP response code: 401 for
URL: http://foobar.oblix.com:80/identity/oblix/apps/
userservcenter/bin/userservcenter.cgi
at sun.net.www.protocol.http.HttpURLConnection.getInputStream
(HttpURLConnection.java:604)
at com.oblix.soapclient.OblixSoapClient.doRequest
(OblixSoapClient.java:285)
```

**Solution**—Make sure that the WebGateStatic.lst file, which is in  
\$WebGate\_install\_dir/access/oblix/apps/webgate directory, has the following  
line:

```
IPValidation:false
```

**16. Problem**—I get the following exception:

```
"com.ibm.ejs.exception.InvalidUserRegistryConfigurationException: User
[username] not authenticated"?
```

**Solution**—Make sure the OB\_UserAttr, OB\_UserSearchAttr and  
OB\_GroupSearchAttr are set correctly in NetPointWASRegistry.properties.

#OB\_UserAttr should be the Login Attribute example LoginID which is uid or  
genuserid

```
OB_UserAttr=samaccountname
```

```
OB_UserSearchAttr=cn
```

```
OB_GroupSearchAttr=cn
```

**17. Problem**—I get the following error in the NetPoint log file:

```
java.lang.StringIndexOutOfBoundsException: String index out of
range: -10 at java.lang.String.substring(String.java(Compiled
Code)) at
com.oblix.soapclient.OblixSoapClient.handleSoapResponse(OblixSo
apClient.java:345) at
com.oblix.soapclient.OblixSoapClient.doRequest(OblixSoapClient.
java:297) at
com.oblix.registry.NetPointWASRegistry.realGetUserDisplayName(N
etPointWASRegistry.java:650) at
com.oblix.registry.NetPointWASRegistry.getUserDisplayName(NetPo
intWASRegistry.java:607) at
com.oblix.registry.NetPointWASRegistry.getUniqueUserId(NetPoint
WASRegistry.java:680) at
com.ibm.ejs.security.registry.CustomRegistryImpl.createCredenti
al(CustomRegistryImpl.java:698) at
com.ibm.ejs.security.registry.CustomRegistryImpl.authenticate(C
ustomRegistryImpl.java:166) at
com.ibm.ejs.security.registry.RegistryBean.authenticate(Registr
yBean.java:109) at
```

Rest of log file:

---

```

com.ibm.ejs.security.registry.EJSRemoteStatelessRegistry.authenticate(EJSRemoteStatelessRegistry.java:25) at
com.ibm.ejs.security.registry._Registry_Stub.authenticate(_Registry_Stub.java:275) at
com.ibm.ejs.security.ltpa.LTPAServerObject.authenticate(LTPAServerObject.java:97) at
com.ibm.ejs.security.util.LTPAAuthenticationCache.update(LTPAAuthenticationCache.java:167) at
com.ibm.ejs.security.util.Cache.get(Cache.java:114) at
com.ibm.ejs.security.util.LTPAAuthenticationCache.getCredential(LTPAAuthenticationCache.java:82) at
com.ibm.ejs.security.SecurityServerBean.authenticateBasicAuthData(SecurityServerBean.java:145) at
com.ibm.ejs.security.EJSRemoteStatelessSecurityServer.authenticateBasicAuthData(EJSRemoteStatelessSecurityServer.j
ava:49) at com.ibm.ejs.security._SecurityServer_Stub.authenticateBasicAuthData(_SecurityServer_Stub.java:281) at
com.ibm.WebSphereSecurityImpl.SecurityServerImpl.authenticateBasicAuthData(SecurityServerImpl.java:69) at
com.ibm.ISecurityLocalObjectLTPAImpl.PrincipalAuthenticatorImpl.authenticate(PrincipalAuthenticatorImpl.java:437)
at com.ibm.ISecurityLocalObjectBaseL13Impl.LoginHelperImpl.request_login_controlled(LoginHelperImpl.java:1092) at
com.ibm.ISecurityLocalObjectBaseL13Impl.LoginHelperImpl.request_login_controlled(LoginHelperImpl.java:827) at
com.ibm.ISecurityLocalObjectBaseL13Impl.CredentialsImpl.get_mapped_credentials(CredentialsImpl.java:1206) at
com.ibm.ISecurityLocalObjectBasicAuthImpl.CredentialsImpl.get_mapped_credentials(CredentialsImpl.java:188) at
com.ibm.ISecurityLocalObjectBaseL13Impl.VaultImpl.setServerCred(VaultImpl.java:3862) at
com.ibm.ISecurityLocalObjectBaseL13Impl.PrincipalAuthenticatorImpl.setServerCred(PrincipalAuthenticatorImpl.java:9
79) at
com.ibm.ISecurityLocalObjectLTPAImpl.PrincipalAuthenticatorImpl.authenticate(PrincipalAuthenticatorImpl.java:302)
at com.ibm.ISecurityLocalObjectBaseL13Impl.LoginHelperImpl.request_login_controlled(LoginHelperImpl.java:1092) at
com.ibm.ISecurityLocalObjectBaseL13Impl.LoginHelperImpl.request_login_controlled(LoginHelperImpl.java:827) at
com.ibm.ISecurityLocalObjectBaseL13Impl.CredentialsImpl.get_mapped_credentials(CredentialsImpl.java:1206) at
com.ibm.ISecurityLocalObjectBasicAuthImpl.CredentialsImpl.get_mapped_credentials(CredentialsImpl.java:188) at
com.ibm.ejs.security.SecurityCollaborator.getActualCredential(SecurityCollaborator.java:999) at
com.ibm.ejs.security.SecurityContext.getActualCreds(SecurityContext.java:75) at
com.ibm.ejs.security.Initializer.bindServerIdToAdminApp(Initializer.java:458) at
com.ibm.ejs.security.Initializer.initialize(Initializer.java:217) at
com.ibm.ejs.security.Initializer.serverStarted(Initializer.java:133) at
com.ibm.ws.runtime.Server.fireServerStarted(Server.java:2001) at
com.ibm.ws.runtime.Server.fireServerStarted(Server.java:1994) at
com.ibm.ejs.sm.server.AdminServer.initializeRuntime0(AdminServer.java:1144) at
com.ibm.ws.runtime.Server.initializeRuntime(Server.java:884) at
com.ibm.ejs.sm.server.AdminServer.main(AdminServer.java:392) at java.lang.reflect.Method.invoke(Native Method) at
com.ibm.ws.bootstrap.WSLauncher.main(WSLauncher.java:158)

```

---

**Solution**—Make sure the COREid Server is up and running.

- 18. Problem**—Why does the Portal Server allow logins with old passwords even though it honors passwords updated through NetPoint?

**Solution**—Because the Portal Server installation sets the Security Cache Timeout to 600 seconds, old passwords will be stored in cache for that amount of time. This parameter is present in the WebSphere Application Server Administrative Console - Security Center and under the General tab.

- 19. Problem**—A deactivated NetPoint user can still access WebSphere resources.

**Solution**—To avoid this, ensure that all authentication schemes that WebSphere uses have the following lines added:

```
(|(! (obuseraccountcontro|=*)) (obuseraccountcontro|=ACTIVATED))
```

- 20. Problem**—Why does the WebSphere Portal Server not come up, resulting in security exceptions?

**Solution**—Ensure that the LPTA Token domain is set up correctly.

- 21. Problem**—After enabling WebSphere security, the Administrative Console does not launch. How can we disable security without disabling the Console?

**Solution**—For each administrative server and administrative agent:

- Change the `com.ibm.CORBA.securityEnabled` property value from `true` to `false` in the `sas.server.props` file that is located in the properties directory; for example, `WAS_install_dir/properties/sas.server.props`, where `WAS_install_dir` is the directory where WebSphere is installed.
- Delete the `sas.server.props.future` file.

#### **On Windows (NT and 2000)**

```
db2cmd
```

```
db2 connect to was40 user <USERID>
```

```
db2 update ejsadmin.securitycfg_table set securityenabled=0
```

```
db2 connect reset
```

#### **On Solaris**

```
source db2cshrc
```

```
db2 list db directory
```

This will show you a list of db2 database instances (may also be called was40).

```
db2 connect to was user <USERID>
```

```
db2 update ejsadmin.securitycfg_table set securityenabled = 0
```

```
db2 commit
```

```
exit
```

Restart the WebSphere Administrative Server.

- 22. Problem**—Single sign-on is not working when a URL resource is protected with a Basic Over LDAP authentication scheme, even though TAI is enabled.

**Solution**—Verify that you followed the steps described in “Configuring the TAI for WebSphere 4 and NetPoint” on page 226 or “Configuring the TAI for WebSphere v5” on page 251, and that you have set the challenge re-direct field in the Basic Over LDAP authentication scheme.

- 23. Problem**—The following error seen in the Event Viewer of the WebSphere Administrative Console:

```

CNTR0020E: Non-application exception occurred while processing
method isAllLevelNone on bean
BeanId(admin#repository.jar#PmiService,
null):javax.transaction.TransactionRolledbackException: CORBA
TRANSACTION_ROLLEDBACK 0 No; nested exception is:
org.omg.CORBA.TRANSACTION_ROLLEDBACK: minor code: 0 completed:
No
org.omg.CORBA.TRANSACTION_ROLLEDBACK: minor code: 0 completed:
No
at
com.ibm.ejs.jts.jts.JBrokerSupport$RI.client_unmarshalled_reque
st (JBrokerSupport.java:405)
at com.ibm.CORBA.iiop.RIs.iterateClientRequestPreRIs(RIs.java
(Compiled Code))
at com.ibm.CORBA.iiop.ClientRequestImpl.reInvoke
(ClientRequestImpl.java:851)
at com.ibm.CORBA.iiop.ClientDelegate.invoke
(ClientDelegate.java:894)
at com.ibm.CORBA.iiop.ClientDelegate.invoke
(ClientDelegate.java:409)
at org.omg.CORBA.portable.ObjectImpl._invoke
(ObjectImpl.java:258)
at com.ibm.ejs.sm.agent._AdminAgent_Stub.invokeActiveObject
(_AdminAgent_Stub.java:39)

```

**Solution**—This may occur if the WebSphere Application Server startup time is long. There can be multiple reasons for this problem, including a startup servlet (load-on-startup = true) which requires long time performing the init() method for the servlet.

If the Ping Initial Timeout is set to a value lower than the amount of time needed for the App Server to start, the Ping Initial Timeout alarm expires before the App Server could come up fully and send the "serverIsAlive" message. As a result, the administration server tries to kill and restart the Application Server process. In this situation, the state of the clone is recorded as Running.

The PMI client indicates that the clone is up and running and tries to invoke the "isAllLevelNone" method on the clone. Because the clone does not exist, it fails with the above error message.

To correct this, set the Ping Initial Timeout to a larger number to allow the Application Server to start completely.

24. **Problem**—After enabling security and configuring LDAP as the Authentication Mechanism, the administration server restarts, the following errors show in the trace file:

```

[02.03.21 08:37:59:957 CST] 4be2cc Initializer W SECJ0007E:
Error during security
initializationjava.lang.NullPointerException at
com.ibm.ejs.security.ltpa.LTPAPrivateKey.decode(LTPAPrivateKey.
java:50) at
com.ibm.ejs.security.ltpa.LTPAPrivateKey.<init>(LTPAPrivateKey.
java:40) at
com.ibm.ejs.security.ltpa.LTPAServerBean.updateAll(LTPAServerBe
an.java:106) at
com.ibm.ejs.security.Initializer.updateActiveLtpaConfig(Initial
izer.java:392) at
com.ibm.ejs.security.Initializer.propagateSecurityConfig(Initia
lizer.java:296) at
com.ibm.ejs.security.Initializer.initialize(Initializer.java:17
3) at
com.ibm.ejs.security.Initializer.serverStarted(Initializer.java
:129) at
com.ibm.ws.runtime.Server.fireServerStarted(Server.java:1977)
at
com.ibm.ws.runtime.Server.fireServerStarted(Server.java:1970)
at
com.ibm.ejs.sm.server.AdminServer.initializeRuntime0(AdminServe
r.java:1123) at
com.ibm.ws.runtime.Server.initializeRuntime(Server.java:882) at
com.ibm.ejs.sm.server.AdminServer.main(AdminServer.java:391) at
java.lang.reflect.Method.invoke(Native Method) at
com.ibm.ws.bootstrap.WSLauncher.main(WSLauncher.java:158) [02.03
.21 08:38:00:001 CST] 4be2cc Server A WSVR0023I: Server
__adminServer open for e-business

```

**Solution**—Sometimes the key and password get out of sync. Create a new key for WebSphere’s use.

To create a new key, do as follows:

- a) From the Security Center (4.0x) or the Global Security Wizard (3.5x) under the Authentication Tab, click Generate Keys.
- b) At the prompt, enter a password for the new key.

**25. Problem**—A user’s attempt to log into the WebSphere Administrative Console results in the following error:

```
ADGU2009E: Security Error: Either username/password is wrong or
this user is not authorized to connect to admin server.
```

**Solution**—Only the following users can log into the WebSphere Administrative Console when security is enabled:

- User defined in the security ID of the custom registry/LDAP.
- Users defined in Administrative roles.

**26. Problem**—I get a Not Authorized error when I try to access a WAS resource or a WebSphere Portal resource.

**Solution**—You get this error because the ObSSOCookie is not being sent. Refresh the page to send the ObSSOCookie.

It is recommended that you use a form-based authentication scheme to avoid this problem. If you use the basic authentication scheme, set the Challenge Redirect field to another WebGate to ensure that the ObSSOCookie gets sent.

**27. Problem**—If TAI is failing with a stack trace:

```
... 1493ff35 JaasLoginHelp E SECJ4001E: Login failed for
testeisintuser/Default Realm ....
```

You get this error because the ObSSOCookie is not being sent.

**Solution**—

a) Refresh the page to send the ObSSOCookie.

b) Enable further security debugging for the following classes:

```
com.ibm.ws.security.*
com.ibm.websphere.security.*
com.ibm.webSphereSecurityImpl.*
SASRas
```

c) To view detailed information on the runtime behavior of security, enable trace on the following components and review the output:

```
com.ibm.ws.security.*=all=enabled:com.ibm.webSphereSecurityImpl.*=all=enabled:c
om.ibm.websphere.security.*=all=enabled.
```

This trace statement collects the trace for the security runtime.

```
com.ibm.ws.console.security.*=all=enabled.
```

This trace statement collects the trace for the security center GUI.

```
SASRas=all=enabled.
```

This trace statement collects the trace for SAS (low-level authentication logic).

The logs should give better debugging messages, and ideas on what exactly is failing.

**28. Problem**—Error in the TAI logs:

Error no action found

**Solution**—Ensure that the Authentication Scheme level for the Authentication scheme protecting the WebSphere Policy (Authen/Basic) is less than or equal to the Authentication scheme level protecting the WebSphere resource.

Ensure that the WAS\_REGISTRY action is set properly (see “Defining a Policy Domain for WebSphere” on page 201). With NetPoint 7.0, ensure that there is an Authorization Expression set and that upon Authorization success the WAS\_REGISTRY action is set.

**29. Problem**—WAS Registrytester.bat may fail:

**Cause**—System variables may be picking up an older version of obaccess.dll based on the path.

**Solution**—Check your system variables to ensure these are correctly set. For example, \$PATH and \$CLASSPATH must be correctly set.

**30. Problem**—Unable to search users in the WebSphere Portal Admin page:

- a) Log in as wpsadmin and go to Portal Administration > Security > Access Control List.
- b) Click Get groups and users. Search for users using the wildcard character “\*” and select wpsadmin and add it to the list, then click OK.
- c) Select user groups in the Select the objects for the permissions for the user wpsadmin, then click Go.
- d) Give Manage permissions for the group All authenticated users for user wpsadmin, then click Save.
- e) Go to Users and Groups > Manage Users, then search for users using the wildcard character “\*.” to display all the users.
- f) To view groups, repeat steps a, b, and c, then give Manage permissions for each group that needs to be viewed for wpsadmin and click Save

**31. Problem**—IBM WebSphere Portal Server gives an error “No Portlets to display”:

- a) Log in to WebSphere Administrative Console. Navigate to WebSphere Portal application.
- b) Click the JVM settings tab and add the following to the System Properties”  
“HttpSession.RecurseThroughProxy”, value = “true”
- c) Restart the WebSphere Portal application.

# Troubleshooting the NetPoint Connector for Portal Server v5

The following Portal Server v5 issues are covered:

- “Portal Server v5 Installation-Related Issues” on page 306
- “Custom Security Integration Related Issues” on page 308

## Portal Server v5 Installation-Related Issues

There are several Portal Server installation-related issues that you may encounter. See details below for more information.

1. **Problem**—Both the WebSphere AppServer and Portal Server installed successfully but not able to access Portal Server page i.e. wps/portal

**Solution**—Check for the fix packs needs to be applied on Application Server for the respective Portal Server version integration. The order to apply fix packs is important.

For Example, In case of Application Server 5.0 and Portal Server 5.0, the Fix Pack 1 needs to be applied on WebSphere Application server. The order of applying patch would be,

- Wasfp1
- Pmefp1
- Fixes
- Manualfixes

You can check `$WAS_install_dir/properties/version/history` folder to confirm the list of fixes applied on Application server.

For more information on this, see the WebSphere Portal infocenter document available on the IBM WebSphere site.

2. **Problem**—On login to the Portal Server (wps/portal), the following error message displayed:

There has been an application error!  
Please contact to your system administrator to report this error.

**Solution**—Please ensure the required Fix Pack has been applied on the Application Server. See 1, above. Also, ensure the Java 2 security is disabled in the WebSphere Application server.

### To disable Java 2 security

- a) Login into WebSphere Application Server Console > Security > Global Security.

- b) Disable the check box for the Enforce Java 2 Security option.
- c) Save the changes, then restart both the WebSphere Application Server and Portal Server.

---

**Note:** It is recommended that you install Portal Server with disabling Global Security in WebSphere Application server.

---

3. **Problem**—On successful login to the Portal Server, the Portal Admin is not able to view the Administration link to manage the Portal Server. This link is displayed on the top-right side of the page.

The Administration portlet is not deployed on the system.

**Solution**—

- a) Run `WPSconfig.{bat | sh} portlets` script to deploy the same.  
This script is available in `$WPS_install_dir/config` folder. For more information check the `wpsconfig.xml` file.

- b) Restart the Portal Server and login using admin credentials.

The Portal Admin user will be able to see the “Administration” link and he can setup the Portal Server Application.

---

**Note:** The same script will also deploy some sample portlet applications come along with Portal Server installation. Also, `WPS_install_dir` is the directory where the Portal Server is installed.

---

4. **Problem**—On successful login to the Portal Server, Portal Admin is not able to view a portlet page. The error message appears as:

There is no content available.

Please check if there is content defined for the markup of your client device.

This message comes because portal page is not created in the Portal Server.

**Solution**—A page has to be created in the Portal Server so you can add portlets on this page.

To add a page, you need page addition/modification privileges and a portlet needs to be installed that adds a new page or edits a page.

See issue 3 to check whether the required portlet has been deployed on the system.

5. **Problem**—When searching users from the Portal Server “user search” interface you may get empty records as a search result. This problem occurs when user names stored in the directory server are in Latin ISO-8859-1 style.

**Solution**—Ensure that you unset the following environment variables.

```
unset LC_COLLATE LC_CTYPE LC_MESSAGES LC_MONETARY LC_NUMERIC
LC_TIME
```

## Custom Security Integration Related Issues

The following issues are related to custom security integration.

1. **Problem**—Portal Server configuration has been done for custom member repository (Netpoint WAS Connector) but still not able to login into the Portal Server using Custom Data Store user id, i.e. user present in Netpoint.

**Solution**—Check `$WPS_install_dir\shared\app\wmm\wmm.xml` file. Confirm that ProfileRepository tag has been configured correctly for Custom Member Repository implementation.

Sometimes WPSconfig script execution modifies wmm.xml file and LDAP settings replace custom member repository settings.

---

**Note:** Oblix recommends that you always keep backup of wmm.xml file configured for Custom Member Repository.

---

2. **Problem**—Portal Admin is able to search for the user present in Custom Data Store but on assigning him a portlet access permission, that user is not able to view the portlet after successful login.

This problem occurs when user DN present in the Custom Data Store contains intermediate spaces.

For Example, cn=Portal User, o=company, c=us.

WebSphere Portal does normalization of DN before matching it with the allowed user DN present in its internal cloudscape database. As string entries do not match, access permissions to the user fails.

**Solution**—To overcome this problem, apply Fix Pack PQ93461 provided by IBM.

3. **Problem**—How can I get/retrieve only first 10 entries (Or number of entries I want to retrieve) on search for NetPoint user/group through the Portal Server.

**Solution**—The parameter of number of records to retrieve is configurable in NetpointWASRegistry.properties file.

The name of parameter is OB\_WebPassXPIRecordsReturned.

If this parameter is not defined or set to zero then it will retrieve all the user/group present in Netpoint Repository.

4. **Problem**—Portal Admin user is not able to install a portlet.

This problem is related to invalid deployment credentials in Portal Server.

**Solution**—Verify that WPSConfig.{bat/sh}  
action-create-deployment-credentials script has been executed or not.

This script execution creates required Credential Vaults in portal server.

Before execution of this script, verify wpsconfig.properties file. Check values of all the configuration parameters mentioned in the Installation Guide.

Also verify that `$WPS_install_dir\shared\app\config\services\VaultService.properties` contains correct `systemcred.dn` value. This should be Portal Admin User DN.

After successful execution verify that the credentials vaults has been created for the Portal Server.

You can check the same from Portal Server Login > Administration > Access > Credential Vault > Manage System Vault Slots.

5. **Problem**—How can I get the debug logs of Netpoint WAS Registry and Netpoint CMR?

**Solution**—For Netpoint WAS Registry logs, set the following parameters of `NetpointWasRegistry.properties` file.

`OB_LogLevel=debug`

`OB_LogFileName=log file complete path`

For enabling NetPoint CMR logs and Portal Server logs set the following parameters in `$WPS_install_dir\shared\app\config\log.properties` file.

`traceString=*=all=enabled`

`com.ibm.wps.services.puma.*=all=enabled:`

`com.ibm.wps.puma.*=all=enabled:`

`com.ibm.wps.command.puma.*=all=enabled:`

`com.ibm.wps.engine.commands.*=all=enabled:`

`com.ibm.wps.services.authentication.*=all=enabled:`

`com.ibm.ws.security.*=all=enabled:`

`com.ibm.websphere.security.*=all=enabled`

All the puma services logs will be generated in the “Trace.log” file.

Refer Portal Server infocenter document for more details on turning on logs.

6. **Problem**—How can I stop/start WebSphere Application Server and Portal Server?

**Solution**—

## To start and stop Application Server use

- Run the startServer command, as shown below:

```
$WAS_install_dir/bin/startServer name_of_app_server
```

(Default name is server1)

- Run the stopServer command, as shown below

```
$WAS_install_dir/bin/stopServer name_of_app_server -username  
WAS_Admin_userid -password WAS_Admin_Pwd
```

## To start and stop Portal Server use

- Run the startServer command, as shown below:

```
$WAS_install_dir/bin/startServer name_of_portal_server
```

(Default name is WebSphere\_Portal)

- Run the stopServer command, as shown below.

```
$WAS_install_dir/bin/stopServer name_of_portal_server -username  
PortalAdmin -password PortalAdminPwd
```

7. **Problem**—Portal Admin is not able to search the users present in NetPoint Repository but these users are able to log in into the Portal Server.

**Solution**—Check the user.fbadefault.filter parameter in the PumaService.properties file. This parameter should contain the attribute name, which is passed to the custom CMR implementation. The NetPoint WAS Connector uses this attribute to ensure the Portal Admin user is a valid user.

Check the trace.log file for exception details.

8. **Problem**—While starting the Portal Server, Netpoint CMR gets invalid Portal Admin credentials.

**Solution**—Execute “Wpconfig. {bat/sh} action-secure-portal-ldap” script. Fire this command from *\$WPS\_install\_dir/config* folder.

9. **Problem**—Why am I unable to install a portlet using the administrator id with NetPoint security enabled?

**Solution**—Complete the following steps to ensure that the proper credentials are being used.

- a) Verify that the correct values for WasUserid and WasPassword are present in the following file:

```
wpconfig.properties
```

WasUserid must be the administrative user id and not the administrative DN. As necessary, use any plain-text editor to open wpconfig.properties and correct the values for WasUserid and WasPassword. Save and close wpconfig.properties.

- b) Confirm that you have executed the following command:

```
wpconfig.bat\.sh action-create-deployment-credentials
```

If you discover that you have not run the command, execute it, then proceed with portlet installation.

If you discover that you previously ran the script using incorrect values for WasUserid or WasPassword, correct the invalid values by executing the following two commands:

```
wpconfig.bat\.sh action-remove-deployment-credentials
```

```
wpconfig.bat\.sh action-create-deployment-credentials
```

- c) Verify that the administrator id is defined correctly by navigating to Portal Administration > Access > Credential Vaults > Manage system vault slots > deployment.user.Vault.

- 10. Problem**—For WebSphere 6.0, no users are returned when a search is conducted through the WebSphere Application Server Administration Console.

Typical, the Netpoint log will contain an error message that begins with the following:

```
Error making SOAP request . . .
```

**Solution**—Make sure that \$LANG and all \$LC\_\* variables are set to en-US. You can check the current values of these variables through the “locale” command on SLES 9.

- 11. Problem**—The Client Cert Authentication feature for the WebSphere Application Server 6.0 is unable to access the Snoop applet on the https port of the Web server.

**Solution**—Ensure that the SSL port used by the Web server has been added to the “default\_host” virtual host configuration on the Application Server. If it does not exist, perform the following steps:

- a) Launch the WebSphere Administration Console
- b) Navigate to Environment > Virtual Hosts > default\_host > host aliases
- c) Add the hostname \* Port : 443 (This is the SSL port used by the Web server)

- d) Save your changes.
- e) Regenerate the plugin by completing the following steps:
- Navigate to Servers > Web servers
  - Select the Web server for which you will create the plugin
  - Select Generate Plugin
- You can view the generated plugin by navigating to: *WebServerName* > plugin properties > View > *PluginFileName*.
- Restart the Web server
- f) To verify that the SSL port is operational, complete the following steps:
- Disable the NetPoint policy protecting the test resource.
  - Attempt to access the test resource using WebSphere Authentication exclusively. Use the following URL:

`https://WebServerHostMachineName:sslPortNumber/snoop`

- If this succeeds, re-enable the NetPoint policy.

# 5 Integrating Oracle Application Servers with NetPoint SSO

This chapter describes the integration of NetPoint 7.0.2 with two Oracle Application Servers: Oracle9i Application Server (Oracle9iAS) and Oracle Application Server 10g (OracleAS 10g).

This chapter covers the following topics:

- “About the NetPoint and Oracle Integration” on page 314
- “Integration Architecture” on page 315
- “Preparing Your Environment” on page 317
- “Integrating NetPoint with Oracle9iAS” on page 318
- “Integrating NetPoint with OracleAS 10g” on page 323
- “Sample Files” on page 332
- “Troubleshooting the Oracle Integration” on page 341

# About the NetPoint and Oracle Integration

Integrating NetPoint with Oracle Application Servers allows the use of NetPoint single sign-on (SSO) and identity management functionality across Web-based applications. This includes applications running on Oracle Application Servers such as Oracle eBusiness Suite, Oracle Forms, Portals, and other NetPoint-protected resources.

## NetPoint Features for Oracle Application Servers

NetPoint provides authentication and single sign-on (SSO) for Oracle9iAS and OracleAS 10g using the following NetPoint authentication schemes:

- Form based
- Basic
- Custom
- Integrated Windows Authentication
- Microsoft .Net Passport

This enables you to use a single user name and password (and optionally a realm ID), to log in to all features of the Oracle Application Servers and other Web applications.

---

**Note:** The Oracle mod\_osso module does not support existing Apache SSL WebGates. Therefore, the Oracle HTTP Server cannot be used for the NetPoint X509 authentication scheme.

---

## Oracle9iAS and OracleAS 10g Infrastructure

Oracle9iAS and OracleAS 10g applications provide a similar infrastructure and a security framework that supports single sign-on for Oracle and other partner applications. The components listed below are necessary to integrate NetPoint and Oracle9iAS and OracleAS 10g. The integration of NetPoint SSO with Oracle9iAS and OracleAS 10g involves the following components.

**Oracle SSO Server**—This provides an SSO framework for Oracle applications and can be configured to accept authentication from third party providers such as NetPoint. This enables SSO integration between NetPoint-protected applications and those applications protected within the Oracle9iAS and OracleAS 10g SSO framework. You can use a single user name and password, and optionally a realm ID, to log in to all features of the Oracle Application server and other Web applications.

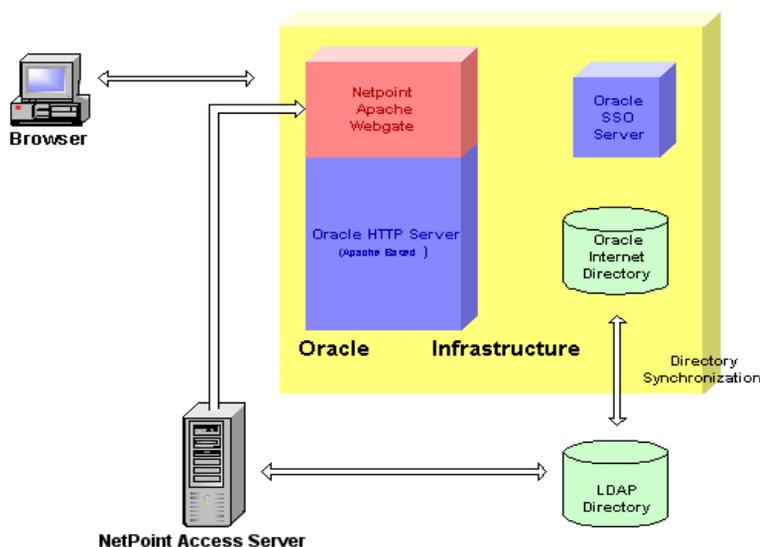
**Oracle HTTP Server (OHS)**—This component of Oracle9iAS and OracleAS 10g provides the Web server interface. OHS is the integration point between NetPoint and Oracle9iAS and OracleAS 10g. During the installation, a NetPoint WebGate will be installed as a module on OHS.

**Oracle Internet Directory (OID)**—Oracle9iAS and OracleAS 10g applications use the Oracle Internet Directory (OID) as the user repository. This is a standards-based LDAP directory which is a component of the management and security infrastructure. The OID can be synchronized with other connected directories.

## Integration Architecture

Figure 12 illustrates the integration between NetPoint and Oracle Application Servers.

**Figure 12** NetPoint and Oracle Integration Architecture



### Process overview: NetPoint with Oracle Integration

1. When a user attempts to access a NetPoint-protected Oracle application or Web resource, the NetPoint WebGate intercepts the request.
2. WebGate requests the security policy from the Access Server to determine if the resource is protected.
3. When the resource is protected, WebGate prompts the user to authenticate.
4. The credentials entered by the user are validated against the directory for authentication.
5. When authentication is successful, an encrypted NetPoint single sign-on cookie is set on the user's browser.

6. After successful authentication, NetPoint determines if the user is authorized by applying policies that have been configured for the resource.
7. Upon successful authorization, NetPoint executes the actions that have been defined in the security policy and sets an HTTP header variable that maps to the Oracle9iAS or OracleAS 10g user ID.
8. The Oracle SSO Server recognizes the NetPoint HeaderVar, authenticates the user, and sets the Oracle SSO Cookie.

---

**Note:** The OID must be synchronized with the NetPoint LDAP directory to ensure that user data is up-to-date. The task of synchronizing OID with the NetPoint LDAP directory is performed by Oracle.

---

## Supported Versions and Platforms

NetPoint 7 supports integration with Oracle as described below.

Table 14 identifies NetPoint support for Oracle Application Server 10g:

**Table 14** COREid Support for Oracle Application Server 10g Rel2(v10.1.2.0.0) on Various Operating System Platforms

COREid Version	Operating System Platforms
7.0.2	Solaris 8 or 9 Windows HP-UX 11.11 (using NetPoint 6.1.1 WebGate only)
7.0.4	Solaris 8 or 9 Windows HP-UX 11.11 (using NetPoint 6.1.1 WebGate only) Red Hat Linux AS 3.0 Windows 2000 or 2003

Table 15 identifies NetPoint support for Oracle9iAS:

**Table 15** NetPoint 7.0 Support for Oracle9iAS

Oracle9iAS	Platforms
Oracle9iAS R2 (SSO) v9.0.2, v9.0.3	Solaris 8 or 9 HP-UX 11.11 (using NetPoint 6.1.1 WebGate only)

# Preparing Your Environment

Before you can integrate NetPoint with Oracle, you must complete the following tasks.

## Task overview: Preparing your Environment

1. Install Oracle9iAS *or* OracleAS 10g, as described in the Oracle installation guide for your specific platform.
2. Install the Oracle Infrastructure, as described in your Oracle documentation:

### Oracle9iAS includes:

- Oracle HTTP Server
- Oracle SSO Server
- Oracle OID

### OracleAS Infrastructure 10g includes:

- Oracle Application Server Metadata Repository
- Oracle SSO Server
- Oracle Internet Directory (a lightweight directory access protocol (LDAP))

---

**Important:** Ensure that the servers on which Oracle and NetPoint are installed have fully-qualified domain names. For example, *hostname.domain.net*.

---

3. Install and set up NetPoint components, as described in the *NetPoint 7.0 Installation Guide*:
  - COREid Server
  - WebPass
  - Access Server
  - Access Manager
4. On the Oracle HTTP Server, install a WebGate for Apache for use with OracleAS 10g and update the Web server configuration file:
  - **Automatic Web Server Updates**—Click Yes to automatically update your Web server configuration file (OHS httpd.conf) during WebGate installation, as described in the *NetPoint 7.0 Installation Guide*.
  - **Manual Web Server Updates**—Use *one* of the methods below:
    - Either*—Locate the OHS httpd.conf file after WebGate installation, add the WebGate entry at the end of the file, then run the following commands on an infrastructure terminal:

```
dcmctl -updateconfig -v
dcmctl restart
```

*Or*—Use the Oracle Enterprise Manager Console to:

Launch the Oracle Enterprise Manager.  
Select the Oracle Application Server hosting the Oracle Infrastructure.  
Select the HTTP Server hosting the WebGate.  
Navigate to Advanced Server Properties.  
From the list of configured files, select httpd.conf for update.  
Include the WebGate entry at the end of the file.

5. Restart the OHS after the Web Server configuration file update.
6. Configure Oracle SSO for external authentication.
7. Configure the Web browser to allow cookies.
8. Proceed as appropriate for your environment:
  - “Integrating NetPoint with Oracle9iAS” on page 318
  - “Integrating NetPoint with OracleAS 10g” on page 323

## Integrating NetPoint with Oracle9iAS

Setting up NetPoint SSO for Oracle9iAS involves configuring Oracle9iAS for external authentication, configuring NetPoint logout, and configuring the Oracle HTTP server for SSL.

### **Task overview: Integrating NetPoint with Oracle9iAS includes**

1. “Preparing Your Environment” on page 317
1. “Configuring Oracle9iAS for Integration with NetPoint” on page 319
2. “Configuring NetPoint for Integration with Oracle9iAS” on page 320
3. “Configuring Logout” on page 321
4. “Testing NetPoint Integration with Oracle” on page 322

# Configuring Oracle 9iAS for Integration with NetPoint

The overall task of preparing Oracle9iAS for integration with NetPoint includes the activities below.

## Task overview: Configuring Oracle 9iAS

1. Edit the Oracle Data Access Descriptors (DADS) file to add the descriptor for NetPoint HeaderVar, as described in “Adding the Descriptor for NetPoint HeaderVar” on page 319.
2. Complete “Configuring the Oracle SSO Server” on page 320.
3. Use the Oracle synchronization tool to synchronize user information between the Oracle OID and the NetPoint LDAP directory.
4. Proceed to “Configuring Logout” on page 321.

---

**Note:** To test the integration without synchronizing the directories, create an Oracle administrator (*orcladmin*) in NetPoint for login purposes.

---

## Adding the Descriptor for NetPoint HeaderVar

In this task, you use the Oracle Enterprise Manager and append a new Plsql descriptor for the NetPoint HeaderVar to the `dads.conf` file.

### To set up Oracle for integration with NetPoint

1. Launch the Oracle Enterprise Manager.
2. Select the Oracle Application Server instance where the Oracle Infrastructure is installed.
3. Select the HTTP Server where WebGate is installed and navigate to Advanced Server Properties.
4. From the list of configured files, select the `dads.conf` file.

The `dads.conf` file opens. This file contains Oracle data access descriptors. You must add a new Plsql descriptor for the NetPoint HeaderVar.

5. In the `dads.conf` file, append the following Plsql descriptor to the list of Plsql descriptors:  

```
PlsqlCGIEnvironmentList REMOTE_USER
```
6. Click Apply.
7. Restart the Oracle HTTP Server.
8. Launch `sqlplus` on the machine where the Oracle Infrastructure is installed.
9. Connect to the Oracle Application Server instance and sign in as *orasso*.

10. Execute the Obsso.pkb package from sqlplus.

The Obsso.pkb package contains the default integration configurations that can be customized as needed. See “Obsso.pkb” on page 333 for the default integration configurations.

## Configuring the Oracle SSO Server

Here, you enable the Oracle SSO server to recognize that the HTTP server is running in SSL mode.

### To configure the Oracle SSO server to recognize SSL mode

1. Install the WebGate for Apache\_SSL.
2. From the command line, run the following Oracle script.

```
ssocfg.sh https machineName https port
```

where *machineName* is the fully-qualified machine name such as xyz.oblix.com and *https port* is the port number of the machine.

## Configuring NetPoint for Integration with Oracle 9iAS

After installing NetPoint and a WebGate for Apache for the Oracle HTTP Server, you need to create NetPoint access control policies to protect Oracle resources.

### To set up NetPoint for integration with Oracle 9iAS

1. Install and set up the COREid and Access Systems, as outlined in “Preparing Your Environment” on page 317.
2. On the Oracle HTTP Server, install WebGate for Apache for use with Oracle 9iAS, as described in the *NetPoint 7.0 Installation Guide*.
3. Navigate to the COREid System Console and create an Oracle Administrator (*orcladmin*) user in NetPoint to match the *orcladmin* user who already exists in the Oracle OID, as described in the *NetPoint 7.0 Administration Guide Volume 1*.
4. Launch the NetPoint Access Manager and define a policy domain to protect /pls (the root of Oracle SSO Server), as shown below and described in more detail in the *NetPoint 7.0 Administration Guide Volume 2*:
  - a) Create an Authentication rule to authenticate users.
  - b) Define an Authorization rule.

- c) In the Authorization rule, create an action for Authorization Success as follows:

```
Return Type = HeaderVar  
Name = REMOTE_USER  
Return Attribute = loginAttribute
```

where *loginAttribute* is the attribute configured as the Login semantic type in the NetPoint COREid System.

Upon successful authorization, the value of *loginAttribute* is passed on to Oracle9iAS Server.

- d) In the Authorization rule, allow access to Anyone.  
e) Enable the Authorization rule.  
f) Enable the Policy Domain.

The single sign-on configuration is now complete.

---

**Note:** To use a HeaderVar that is different from REMOTE\_USER, replace REMOTE\_USER with the desired variable in the following three locations: In NetPoint Authorization Rule > Actions, in the Oracle PlsqlCGIEnvironmentList (see “To set up Oracle for integration with NetPoint” on page 319), and in NetPoint Obsso.pkb package (see “Obsso.pkb” on page 333).

---

5. Proceed to “Configuring Logout” on page 321.

## Configuring Logout

You must specify session synchronization such that when a user logs out, the user is logged out of all Oracle-protected and NetPoint-protected resources.

To do this, you must create a logout file that contains the command, which when executed, deletes both the Oracle and NetPoint SSO cookies. This ensures that the user cannot access any NetPoint-protected or Oracle-protected resource without logging in again.

### To configure logout

1. Create a Logout.html file in *\$ORACLE\_HOME*/Apache/Apache/htdocs where *\$ORACLE\_HOME* is the name of the directory where Oracle is installed.

2. From sqlplus, run the following command to set the location of the Logout URL in Oracle:

```
update wvss01s_configuration_info$ set login_url="UNUSED UNUSED  
http://machineName:port/logout.html;  
commit;
```

Where *machineName* is the name of the machine where the Oracle HTTP Server is running and *port* is the port number of the machine.

---

**Note:** You can replace this URL with any URL that contains “logout.”

---

A logout link is displayed when users log in to an Oracle Web resource through NetPoint. When users log out of the resource, NetPoint logs them out of all NetPoint-protected resources.

Any SQL statement or package must be executed as the orasso user. By default the *orasso* password is set during the installation of the Oracle9iAS Infrastructure.

3. Complete the appropriate activities below for your environment:
  - “To conduct an LDAP search for the password” on page 322
  - To continue with “Testing NetPoint Integration with Oracle” on page 322

### **To conduct an LDAP search for the password**

1. On the Infrastructure machine, enter the following command at the command line:

```
ldapsearch -D cn=orcladmin -w superuser_password -p 4032 -h  
hostname -b "cn=IAS,cn=Products,cn=OracleContext" -s sub -v  
OrclresourceName=ORASSO | grep orclpasswordattribute
```

where *superuser\_password* is the password of the orcladmin user of OID. By default, this is the same as your *ias\_admin* user after a new infrastructure install and *hostname* is the name of the machine where the infrastructure is installed.

2. Determine the password.

## **Testing NetPoint Integration with Oracle**

After you set up Oracle and NetPoint for integration, test to ensure that the integration is successful.

### **To test NetPoint SSO for Oracle**

1. Enter the following URL in the browser:  
`http://machineName:port/pls/orasso/`

where *machineName* is the machine where Oracle9iAS Server is installed and *port* is the port number of the machine.

NetPoint challenges you for credentials. After you have successfully authenticated, the Oracle Web resource page appears.

2. Click the Login button.

If NetPoint SSO is successful, you will be allowed access to the page without being challenged for authentication.

3. When you are ready to log out, click the Logout link.

If NetPoint SSO is successful, you will be logged out of all NetPoint-protected resources.

## Integrating NetPoint with OracleAS 10g

The procedures to integrate NetPoint with OracleAS 10g Application Server are similar to, yet more granular than, integration with Oracle9iAS. Each Oracle application's configuration is provided separately. Setting up NetPoint SSO for OracleAS 10g involves configuring OracleAS 10g for external authentication and configuring NetPoint logout.

---

**Note:** Currently there is no Apache SSL WebGate that supports Oracle mod\_ossl. Therefore SSL configuration does not apply here.

---

### Task overview: Integrating NetPoint with OracleAS 10g

1. “Preparing Your Environment” on page 317.
2. “Configuring OracleAS 10g for Integration with NetPoint” on page 323
3. “Configuring NetPoint for Integration with OracleAS 10g” on page 330
4. “Testing NetPoint Integration with Oracle” on page 322 is the same regardless of which Oracle Application Server you are using.

## Configuring OracleAS 10g for Integration with NetPoint

You complete the following procedures to set up OracleAS 10g for the integration:

- “Enabling Single-Sign On” on page 324
- “Integrating the Delegated Administration Service” on page 325
- “Integrating the Portal” on page 326
- “Enabling Single-Sign On for Forms” on page 328
- “Integrating Reports Services” on page 329
- “Synchronizing the OID and NetPoint LDAP Directory” on page 329
- “Implementing Global Logout from SSO and Access Server” on page 330.

## Enabling Single-Sign On

Enabling single-sign on for the integration between NetPoint and OracleAS 10g includes several activities, discussed below:

- “Creating the Java Class for Integration” on page 324.
- “Changing the Policy.properties file” on page 325
- “Testing the SSO implementation” on page 325

### Creating the Java Class for Integration

The first step in enabling SSO for the integration involves coding a JAVA class, which will look for the Header variable from Oblix.

---

**Note:** This example assumes you have installed and set up the COREid System and Access System, created a policy domain in NetPoint, and defined an authorization action that sets a Header Variable with the ID of the user. For details, see “Integrating NetPoint with OracleAS 10g” on page 323.

---

### To code a JAVA class to look for a NetPoint HeaderVar

1. Download the jar file ipastoolkit.jar in the directory below on your Oracle Infrastructure installation.

*ORCALE\_HOME/sso/lib*

2. Copy the source code from the Sample Files section #SSOOblixAuth.java into the editor and save it as SSOOblixAuth.java.
3. Include the downloaded jar file ipastoolkit.jar in your class path, along with servelet.jar.
4. Compile the java file, then copy the class file on to Infrastructure installation, as described below:

- a) At the command prompt, navigate to the directory below on the Infrastructure Installation node.

*ORCALE\_HOME/sso/plugin*

- b) Create the directory structure oblix/security/ssoplugin within the sso/plugin directory.
- c) Copy the SSOOblixAuth.class into the directory below.

*ORCALE\_HOME/sso/plugin/oblix/security/ssoplugin*

## Changing the Policy.properties file

Next you need to change the Policy.properties file to register the Java class for integration. The Policy.properties file can be found in the directory below on the on the Infrastructure Installation node.

```
ORCALE_HOME/sso/conf
```

### To register the Java class

1. In this class, navigate to the line MediumSecurity\_AuthPlugin.
2. Comment out the existing line and add a new line to register your java class, as shown below.

```
MediumSecurity_AuthPlugin = oblix.security.ssoplugin.SSOoblixAuth
```

3. Save the file.
4. Restart the oc4j instance OC4J\_SECURITY to have your changes to take affect:

```
dcmctl -updateconfig -v  
dcmctl restart
```

### Testing the SSO implementation

If SSO is working properly, you are challenged by WebGate when you attempt to access the SSO login URL. Once you enter credentials and they are validated, the Header variable is extracted and you are not challenged for authentication again. This is because of the Java class integration.

### To test the SSO implementation

1. Using a browser, enter the SSO login URL:

```
http://Infra-machine-name:port/pls/orasso
```

The default port is 7777. Always use fully-qualified domain names to access Oracle Applications.

2. Enter your login credentials.

When you have already passed authentication, you should be granted access to the page without having to supply credentials a second time.

## Integrating the Delegated Administration Service

The Delegated Administration Service (DAS) is part of the Oracle Identity Management, an integrated infrastructure that includes the following components:

- **Oracle Internet Directory**—An LDAP V3-compliant directory service

- **Delegated Administration Service (DAS)**—The Oracle Internet Directory component that provides trusted proxy-based administration of directory information by users and application administrators.
- **Oracle Directory Integration Service**—A component of the Oracle Internet Directory that permits synchronization between the Oracle Internet Directory and other directories and user repositories
- **Provisioning Integration Service**—The Oracle Internet Directory component that provides automatic provisioning of services, as described in Oracle documentation

The DAS is installed by default when you install the OracleAS 10g Infrastructure, and should integrate automatically. No additional steps are needed for a user to access DAS when NetPoint is integrated with SSO.

The DAS link is:

*[http:// Infra-machine-name:port/oiddas](http://Infra-machine-name:port/oiddas)*

---

**Note:** If you experience errors using Create/Edit user and Create/Edit groups portlets, move the DAS to the middle tier from the Infrastructure. For details, see “Integrating the Portal” on page 326.

---

## Integrating the Portal

The Oracle Application Server Portal enables you to build, deploy, and maintain self-service, integrated EIPs. A customized portal page can present information from different providers and can include both enterprise search and directory lookup fields.

A portal page consists of multiple portlets. Each portlet is a region of the portal page that provides dynamic access to a Web-based resource.

When NetPoint SSO is integrated with OracleAS 10g, users should be able to access the portal below:

*[http://midtier\\_Home:port/pls/portal](http://midtier_Home:port/pls/portal)*

---

**Note:** The Create/Edit user and Create/Edit groups portlets call the DAS from the portal. If you experience errors using Create/Edit user and Create/Edit groups portlets, you need to move the DAS to the middle tier from the Infrastructure.

---

## To deploy DAS on middle tier

1. On the middle tier, create an OC4J instance named OC4J\_SECURITY using either the Enterprise Manager or the dcmctl command line.
2. Deploy the oiddas.ear file on this OC4J\_SECURITY instance by copying the file from the infrastructure installation as follows:

```
ORACLE_Home/ldap/das
```

3. Deploy the EAR file using one of the methods below:

- Use the Enterprise Manager
- Use the dcmctl command for DAS deployment, as described in the *Oracle Internet Directory Administrators Guide 10G (9.0.4) Part Number B12118-01*, chapter 30, subsection “Configuring Oracle Delegated Administration Services in a New Oracle Home”.

4. Start the component.

```
OC4J_SECURITY
```

5. Login using the cn=orcladmin account, then expand the tree, as shown below:

```
Entry Management
  cn=OracleContext
    cn=Products
      cn=DAS
```

6. Select the cn=OperationURLs page (do *not* expand the tree).

7. Click on the cn=OperationURLs page.

The right hand pane displays a number of fields or attributes. The last field is orcldasurlbase and is set to the Infrastructure HTTP Server and port.

8. Change this entry to reflect the middle tier HTTP Server and port.

9. On the portal, complete the steps below:

- a) Log into Portal as the user orcladmin.
- b) Click the Builder link.
- c) Navigate to the Administer tab > Global Settings link in the center column.
- d) Click the SSO/OID tab.
- e) Select the checkbox beside “Refresh Cache for OID Parameters” at the bottom of the page, then click on the OK button.
- f) Clear the mod\_plsql cache.
- g) Remove all files and directories below:

```
MidTier_Home\Apache\modplsql\cache\plsql
```

```
MidTier_Home\Apache\modplsql\cache\session
```

10. Update the table manually in the portal.
11. Log in to the portal schema and check the DASAPPLICATION column in the table PORTAL.WWPTL\_OTHER\_SERVICES\_LINK\$.
12. If this link is still pointing to the Infrastructure, update the table to point to the middle tier URL.

These changes should point the Portal application to the DAS on the middle tier.

## Enabling Single-Sign On for Forms

The Oracle Application Server Forms Services is a middle-tier application framework that you use to deploy complex transactional forms applications to the internet.

When you integrate NetPoint with OracleAS 10g, you need to enable SSO for forms. Once SSO is enabled for forms, NetPoint handles authentication and you should not be challenged to enter the schema userid/password either by the SSO login page or by the forms.

### To enable SSO for forms

1. Locate the forms90.conf file located in the directory below:

*MidTier\_Home/forms90/server*

2. At the end of the forms90.conf file add the following lines.

```
<IfModule mod_osso.c>
    <Location /forms90/f90servlet>
        require valid-user
        AuthType Basic
    </Location>
</IfModule>
```

3. Restart OC4J\_BI\_FORMS and the forms server to have you changes take affect.

Next you create a Resource Access Descriptor (RAD) for the OID users. A RAD can be created at a global level so all users can use the same RAD to access the resource. Alternatively, the RAD can be created for each user.

4. Create a Resource Access Descriptor (RAD) for the OID users to map the LDAP user to the Database schema.

The next step can be done at the global level in the formsweb.cfg file (the default configuration), or at the application level to make individual applications SSO enabled.

5. Set the ssoMode to true to make the application SSO enabled using the Enterprise Manager to update the formsweb.cfg file.

For example, to make an individual application SSO enabled:

```
[myApp]
form=myFmxs
ssoMode=true
```

For more information, see chapter 6 in the *Oracle Application Server Forms Services Deployment Guide 10G (9.0.4)* for Windows and Unix, Part No. B10470-02.

6. Test this implementation by navigating to the following URL:

[http://midtier\\_Home:port/forms90/f90servlet?config=default](http://midtier_Home:port/forms90/f90servlet?config=default)

## Integrating Reports Services

The Oracle Application Server Reports Services allow you to deploy reports to the OracleAS 10g, as described in your Oracle documentation.

Reports are SSO-enabled out of the box and should work without further steps when you integrate NetPoint with OracleAS 10g.

### To access the protected reports page

1. Navigate your browser to the URL below:  
<http://machine:port/reports/rwservlet/showenv>
2. Log in when challenged by WebGate.
3. Confirm that once authenticated you can view the Environment settings for Oracle Reports (an SSO-protected page).

For more information, see chapter 10 of the *Oracle Application Server Reports Services Publishing Reports to the Web 10G (9.0.4)*, Part No B13673-01.

## Synchronizing the OID and NetPoint LDAP Directory

The next step in the configuration of OracleAS 10g for integration with NetPoint is to use the Oracle synchronization tool to synchronize user information between the Oracle OID and the LDAP directory server used by NetPoint.

For details about this synchronization tool and process, see your Oracle OID documentation.

---

**Note:** To test the integration without synchronizing the directories, you need to create an Oracle administrator (oracladmin) within NetPoint for login purposes.

---

## Implementing Global Logout from SSO and Access Server

The NetPoint Access Server recognizes the logout URL in the format `logout.*`. As a result, the default SSO logout page does not work with SSO. The discussion “Logout.jsp” on page 338 provides a sample file you need to use for the logout.

### To implement global logout from SSO

1. Copy the sample “Logout.jsp” on page 338 source into a text editor and save it as `logout.jsp`.
2. Copy the `logout.jsp` into the directory below:  
`Infra_Home/j2ee/OC4J_SECURITY/applications/sso/web/jsp`
3. Register the JSP as the logout page with SSO, as follows:
  - a) Login to ORASSO schema.
  - b) Update the `wwsso_ls_configuration_info$` table.
  - c) Update `wwsso_ls_configuration_info$` set `login_url='UNUSED  
UNUSED http://<Infra-machine-name>:<port>/sso/jsp/logout.jsp'`
4. Confirm that you can perform a global logout both from SSO and from the Access Server.

## Configuring NetPoint for Integration with OracleAS 10g

After installing NetPoint, and a WebGate for Apache on the Oracle HTTP Server, you need to create NetPoint access control policies to protect Oracle resources.

### Task overview: Setting up NetPoint for integration with OracleAS 10g includes

1. Install and set up the COREid System and Access System, as outlined in “Preparing Your Environment” on page 317.
2. Navigate to the COREid System Console and create an Oracle Administrator (`orcladmin`) user in NetPoint to match the `orcladmin` user who already exists in the Oracle OID, as described in the *NetPoint 7.0 Administration Guide Volume I*.
3. Complete “Protecting the Single-Sign On Login URL” on page 331.

## Protecting the Single-Sign On Login URL

You need to protect the SSO login URLs below so the WebGate challenges the user whenever the OracleAS 10g SSO is accessed:

```
/pls/orasso/orasso.wvssso_app_admin.ls_login  
/sso/auth/*
```

The following activities are required to protect the SSO login URLs, or any other resources, using NetPoint.

Each step in the task below is a full procedure. For complete details, see the *NetPoint 7.0 Administration Guide Volume 2* as all details are not repeated here.

### Task overview: Protecting resources with NetPoint

1. Define an authentication scheme using the Access System Console.

For example:

```
Access System Console > Access System Configuration > Authentication  
Management > Add
```

2. Create a policy domain using the Access Manager.

For example:

```
Access Manager > Create Policy Domain
```

3. Add a Resource to your policy domain using the Access Manager.

For example:

```
Access Manager > Create Policy Domain > Resources
```

4. Define rules for your policy domain using the Access Manager.

For example:

```
Access Manager > Create Policy Domain > Default Rules
```

5. Define an Authorization action that sets a Header Variable with the ID of the user.

For example:

```
Access Manager > Create Policy Domain > Default Rules > Authorization  
Expressions > Actions
```

```
Authorization Success
```

```
Return
```

```
Type—HeaderVar
```

```
Name—REMOTE_USER
```

```
Return Attribute—loginAttribute
```

where *loginAttribute* is the attribute configured as the Login semantic type in the NetPoint COREid System. This name must map to the login name of the

user stored in the Oracle SSO repository. Some people have used the “EMPLID” attribute, which will pass the Employee ID of logged in user.

Upon successful authorization, the value of *loginAttribute* is passed on to the OracleAS 10G server.

---

**Note:** To use a HeaderVar that is different from REMOTE\_USER, you need to replace REMOTE\_USER with the desired variable in two locations: NetPoint Access System Console > Authorization Rule > Actions, and in the Oracle Java class (see “Creating the Java Class for Integration” on page 324).

---

6. In the Authorization rule, allow access to Anyone.

For example:

Access Manager > Create Policy Domain > Authorization Rules > *Name* > Allow Access > Any one

7. Enable the Authorization rule.

For example:

Access Manager > Create Policy Domain > Authorization Rules > *Name* >

8. Enable the Policy Domain.

For example:

Access Manager > My Policy Domains > *Name* > Modify > Enabled

The single sign-on configuration is now complete.

9. Test your policy domain, as described in the *NetPoint 7.0 Administration Guide Volume 2*.

## Sample Files

Topics here are divided into two groups, as follows:

- “Oracle9iAS Files” on page 332
- “OracleAS 10g Files” on page 336

## Oracle9iAS Files

You can customize the following two sample files to meet your requirements:

- “Obsso.pkb” on page 333
- “Logout Page” on page 334

## Obsso.pkb

The contents of the Obsso.pkb package that is used to integrate NetPoint with Oracle are detailed below. To use the default implementation, create a file as shown below. You can also customize the file as needed.

```
ssooblix.pkb
```

```
Copyright (c) Oracle Corporation 2001. All Rights Reserved.
```

```
NAME
```

```
ssooblix.pkb - Single Sign-On External Auth Module for Oblix  
NetPoint
```

```
DESCRIPTION
```

```
This package body is used to achieve integration with Oblix  
NetPoint. It may be customized as required. This is just a default  
implementation and changes might be required based on customer's  
specific deployment scenario.
```

```
CREATE OR replace PACKAGE BODY wwsso_auth_external AS
```

```
FUNCTION authenticate_user
```

```
(
```

```
    p_user OUT VARCHAR2
```

```
)
```

```
return PLS_INTEGER
```

```
IS
```

```
    l_http_header varchar(1000);
```

```
    l_ssouser wwsec_person.user_name%type := NULL;
```

```
BEGIN
```

```
l_http_header := owa_util.get_cgi_env('REMOTE_USER');
```

```
if l_http_header is not null then
```

```
    l_ssouser := l_http_header;
```

```
else
```

```
    debug_print('OBLIX HEADER is null');
```

```
    raise EXT_AUTH_FAILURE_EXCEPTION;
```

```
end if;
```

```
p_user := NLS_UPPER(l_ssouser);
```

```
    return 0;
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        debug_print('unknown exception in authenticate_user(p_user)'  
            || sqlerrm);
```

```
        RAISE EXT_AUTH_FAILURE_EXCEPTION;
```

```
END authenticate_user;
```

```
FUNCTION get_authentication_name
```

```
RETURN VARCHAR2
```

```
AS
```

```

BEGIN
RETURN 'Oblix Netpoint SSO';

END get_authentication_name;
PROCEDURE set_external_cookies
(
    p_username IN VARCHAR2
    ,p_password IN VARCHAR2
    ,p_cookie_list OUT wwsso_ls_private.cookie_list
)
AS
BEGIN
    null;

END set_external_cookies;
END;
/
show errors

```

## Logout Page

The following is a sample logout.jsp file that is used to create a logout page. To use the default implementation, create a file as shown below. You can also customize it as needed:

```

<html>
<body bgcolor="white">
<%@ page buffer="5" autoFlush="true" %>
<%
response.setHeader("Cache-Control", "no-cache");
response.setHeader("Pragma", "no-cache");
response.setHeader("Expires", "Thu, 29 Oct 1970 17:04:19 GMT");
String done_url = null;
int i = 0;
try
{
    done_url = request.getParameterValues("p_done_url")[0]; %>
<center><h1>Single Sign-Off</h1><p>
<table border=0>
<tr>
<th>Application Name</th>
<th>Logout Status</th>
</tr>

```

```

    <% for(;;)
    {
        i++;
        String app_name =
request.getParameterValues("p_app_name"+i)[0];
        String url_name =
request.getParameterValues("p_app_logout_url"+i)[0]; %>
        <tr>
        <td><%=app_name%></td>
        <td><img src=<%= url_name%>></td>
        </tr>
    <% }
}
catch(Exception e)
{
    if(i>1)
    { %>
        </table>
        <br>
        <form><INPUT TYPE=button NAME=p_request VALUE=Return
onClick=javascript:submitForm()></form></center>
        <%}
    else
    {%>
        <h2><center><font color=red>ERROR:</font>
        This page can not be accessed directly!</center></h2>
        <% }
}

%>
</body>
</html>
<script>
function submitForm()
{
    document.location.href ='<%= done_url %>';
}
</script>

```

## OracleAS 10g Files

The two sample files below can be customized to meet your requirements:

- “SSOblAuth.java” on page 336
- “Logout.jsp” on page 338

### SSOblAuth.java

```
package oblix.security.ssoplugin;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import oracle.security.sso.ias904.toolkit.IPASAuthInterface;
import oracle.security.sso.ias904.toolkit.IPASAuthException;
import oracle.security.sso.ias904.toolkit.IPASUserInfo;
import oracle.security.sso.ias904.toolkit.IPASInsufficientCredException;
import java.net.URL;
import java.util.*;

public class SSOblAuth implements IPASAuthInterface
{
    private static String OBLIX_USER_HEADER = "REMOTE_USER";
    private static String CLASS_NAME = "SSOblAuth";

    public SSOblAuth()
    {
        System.out.println("Inside SSOblAuth constructor.....");
    }
    public IPASUserInfo authenticate(HttpServletRequest request)
        throws IPASAuthException, IPASInsufficientCredException {

        String OblixUserName = null;

        Enumeration en=request.getHeaderNames();
        while(en.hasMoreElements())
        {
            String hr=en.nextElement().toString();
            System.out.println("Header ....."+hr+"
value....."+request.getHeader(hr));
```

```

}

try
{
    System.out.println(".....Getting Header Variable.....");
    OblixUserName = request.getHeader(OBLIX_USER_HEADER);

    System.out.println("The Header name....."+OblixUserName);
}
catch (Exception e)
{
    throw new IPASInsufficientCredException("No Oblix Header");
}

if (OblixUserName == null)
    throw new IPASInsufficientCredException("No Oblix Header");

IPASUserInfo authUser = new IPASUserInfo(OblixUserName);
System.out.println("The IPASUserInfo Class....."+authUser);
return authUser;
}

public URL getUserCredentialPage(HttpServletRequest request,String msg) {

System.out.println("Inside Get User Credential Page .....Should not come
here>.....");

    URL errorURL=null;
    try
    {
        errorURL=new URL(new String(request.getRequestURL()));
    }
    catch(Exception ee){};
    return errorURL;
}

```

```
}
```

## Logout.jsp

You can use the sample file below as discussed in “Implementing Global Logout from SSO and Access Server” on page 330.

```
<!-- Copyright (c) 1999, 2003, Oracle. All rights reserved. -->
```

```
<%@page autoFlush="true" session="false"%>
```

```
<%
```

```
    // Declare English Message Strings
```

```
    String msg1 = "Single Sign-Off";
```

```
    String msg2 = "Application Name";
```

```
    String msg3 = "Logout Status";
```

```
    String msg4 = "ERROR: The return URL value not found.";
```

```
    String msg5 = "ERROR: Logout URL for partner applications not found.";
```

```
    // Get the user language preference
```

```
    String userLocaleParam = null;
```

```
    java.util.Locale myLocale = null;
```

```
    // Get the user locale preference sent by the SSO server
```

```
    try
```

```
    {
```

```
        userLocaleParam = request.getParameterValues("locale")[0];
```

```
    }
```

```
    catch(Exception e)
```

```
    {
```

```
        userLocaleParam = null;
```

```
    }
```

```
    if( (userLocaleParam == null) || userLocaleParam.equals("") )
```

```
    {
```

```
        myLocale = request.getLocale();
```

```
    }
```

```
    else
```

```
    {
```

```
        if(userLocaleParam.indexOf("-") > 0 )
```

```
        {
```

```

        // SSO server sent the language and territory value (e.g. en-us)
        myLocale = new java.util.Locale(
            userLocaleParam.substring(0, 2)
            , userLocaleParam.substring(3, 5));
    }
    else
    {
        // SSO server sent only the language value (e.g. en)
        myLocale = new java.util.Locale(userLocaleParam, "");
    }
}

//The below two lines will be used only for the Multilingual support with //
proper resource bundle class supplied

//java.util.ResourceBundle myMsgBundle
// = java.util.ResourceBundle.getBundle("MyMsgBundleClassName", myLocale);

// Get the message string in the appropriate language using the message key.
// Use this string to display the message in this page.

// String mesg = myMsgBundle.getString("mesg_key");

%>

<html>
<body bgcolor="#FFFFFF">
<h1><%=msg1%></h1>

<%
String done_url = null;
int i = 0;

// Get the return URL value
try
{
    done_url = request.getParameterValues("p_done_url")[0];
}
catch(Exception e)
{

```

```

    done_url = "";
}

// Get the application name and logout URL for each partner application
try
{
%>
<b> <%=msg2%> &nbsp; <%=msg3%> </b>
<br>
<%
    for(;;)
    {
        i++;

        String app_name = request.getParameterValues("p_app_name"+i)[0];
        String url_name = request.getParameterValues("p_app_logout_url"+i)[0];
%>
        <%=app_name%>
        &nbsp;
        
        <br>
<%
    }
}
catch(Exception e)
{
    if(done_url == null)
    {
%>
        <%=msg4%> <br>
<%
    }

    if(i>1)
    {
%>
        <br> <a href="<%=done_url%>">Return</a>
<%
    }
}

```

```

else
{
%>
    <%=msg5%><br>
<%
}
}
%>

</body>
</html>

```

## Troubleshooting the Oracle Integration

This discussion is divided into two parts:

- “Troubleshooting the Oracle9iAS Integration” on page 341
- “Troubleshooting the OracleAS 10g Integration” on page 344.

### Troubleshooting the Oracle9iAS Integration

**Problem**—Oracle is not reading the NetPoint HeaderVar.

**Solution**—Complete the procedure below to ensure that Oracle is reading the HeaderVar.

#### To create the procedure to ensure that Oracle is reading the HeaderVar

1. Create or replace oblix\_snoop as:

```

l_var varchar2(2000) default null;
begin
    l_var := owa_util.get_cgi_env('REMOTE_USER');
    http.p('username: ' || l_var);
end oblix_snoop;
/

```

2. Grant Execute permission on oblix\_snoop to Public and Commit.

3. Run the following URL from the browser:

`http://machineName:port/pls/orasso/orasso.oblix.snoop`

The value of REMOTE\_USER (or an other HeaderVar) will be displayed in the browser window.

**Problem**—WebGate is not installed correctly.

**Solution**—Complete the procedure below to ensure that WebGate is installed correctly.

### **To ensure the WebGate is installed correctly**

1. Create a file named test/index.html.
2. In NetPoint, create a policy domain to protect /test.
3. Attempt to access the URL where /test is located.

If WebGate is installed correctly, you will be challenged, and upon successful authentication, allowed to access /test.

**Problem**—SSO is not working. To debug the wwsso\_auth\_external package, how can I read the output of debug\_print statements in it?

**Solution**—Complete the procedure below.

### **To resolve this issue**

1. Execute the following package from sqlplus with orasso as the user to put the values of debug\_print into the wwsso\_log\$ table.

```
CREATE OR replace PROCEDURE debug_print (str VARCHAR2) AS
PRAGMA autonomous_transaction;
BEGIN
INSERT INTO wwsso_log$ VALUES
    (wwsso_log_pk_seq.nextval,
    substr(str, 1, 1000),
    sysdate,
    dbms_session.unique_session_id
    );
commit;
END debug_print;
```

2. Try to access any resource under /pls to execute wwsso\_auth\_external.
3. From sqlplus, run the following statement below:

```
select * from wwsso_log$;
```

**Problem**—Using a form-based authentication scheme while accessing OIDDAS, Form application, or externally deployed J2ee applications, the Oracle SSO login page is displayed after the NetPoint form login page.

In these applications, mod\_osso uses a POST based redirection method instead of GET to call the SSO server. The redirection method used is based on the value of OssoRedirectByForm directive. To use the GET method, the directive should be set to false.

---

**Note:** The OssoRedirectByForm directive is present starting with the v9.0.2.3 patch set. If your patch set is earlier, you need to perform a patch upgrade to v9.0.2.3.

---

**Solution**—Complete the procedure below to verify the value of OssoRedirectByForm directive:

### To verify value of OssoRedirectByForm directive

1. Launch the Oracle Enterprise Manager.
2. Select the Oracle Application Server instance where the Oracle Infrastructure is installed.
3. Select the HTTP Server where WebGate is installed and navigate to Advanced Server Properties.
4. From the list of configured files, select the mod\_osso.conf file.
5. Check to see if OssoRedirectByForm is set to false (by default in 9.0.2.X code, it defaults to true).
6. If the default directive value is used, set it to false as highlighted in **bold** below:

```
<IfModule mod_osso.c>
ossoIpcheck off
ossoIdleTimeout off
ossoConfigFile
/private1/iasinst/install_set1/904infra/Apache/Apache/conf/
osso/osso.conf
OssoRedirectByForm off
</IfModule>
```
7. Click Apply.
8. Restart the Oracle HTTP Server.

## Troubleshooting the OracleAS 10g Integration

The following tips are specific to the Oracle 10g integration. For more tips, see “Troubleshooting the Oracle9iAS Integration” on page 341.

**Problem:** With a form-based authentication scheme, while accessing OIDDAS/ Form application/ externally deployed j2ee applications, the Oracle SSO login page is displayed after the NetPoint Form login page.

**Solution:** This happens if mod\_osso uses a POST based redirection method instead of GET to call the SSO server. The redirection method used is based on value of OsoRedirectByForm directive. To use GET method, this directive needs to be set to false. In Oracle 10G Application Server, this value is set to false by default.

### To verify that this directive is set to false

1. Verify the value of OsoRedirectByForm directive.
2. Launch the Oracle Enterprise Manager.
3. Select the Oracle Application Server instance where the Oracle Infrastructure is installed.
4. Select the HTTP Server where WebGate is installed and navigate to Advanced Server Properties.
5. From the list of configured files, select the mod\_osso.conf file.
6. Check if OsoRedirectByForm is set to true. By default the values is false.
7. If default directive value is not used, set it to false as highlighted below

```
<IfModule mod_osso.c>
OsoIpCheck off
OsoIdleTimeout off
OsoConfigFile
/private1/iasinst/install_set1/904infra/Apache/Apache/conf/
osso/osso.conf
OsoRedirectByForm off
</IfModule>
```
8. Click Apply.
9. Restart the Oracle HTTP Server.

**Problem:** How do I find ORASSO and Portal schema passwords?

**Solution:** Complete the steps below.

### **To find these database schema passwords**

1. Login to Oracle Directory Manager as super user 'orcladmin'.
2. Expand the tree on the left hand side, as shown below:

Cn= OracleContext

    Cn=Products

        Cn=IAS

            Cn=IAS Infrastructure Databases

                OrclReferenceName=<global database name>

                    OrclResourceName=ORASSO

3. Click on the ORASSO entry and look for the value for attribute 'orclpasswordattribute' (the Password for ORASSO schema).

---

**Note:** Similarly you can click on the OrclResourceName=PORTAL for the portal schema password.

---

**Problem:** How do I check the SSO logs?

**Solution:** You can view the SSO logs from Enterprise Manager (EM).

1. Logon to EM, click on the 'Logs' link at the bottom.  
    You will get a search screen.
2. From the 'Available Components' list select 'Single Sign-on:orasso' and move it to the 'Selected Components'.
3. Now perform the search to view the SSO logs.

**Problem:** How do I create a default RAD?

**Solution:** Complete the steps below to create a default RAD:

### **To create a default RAD**

1. Access OIDDAS Console > Configuration > Preference, as usual.
2. Scroll to the bottom of the page to display "Resource Access Information".
3. Click "Create" to create a new resource file.

4. Enter a Resource Name:

For example, for a default configuration you can use:

*default*

---

**Note:** Resource name created over here should be the same as the configuration present in formsweb.cfg file.

---

5. Click Next and fill in the userid/password and the connect string for the database, click Submit.

The Username is a valid DB user. Database refers to the DB used. Therefore, if a schema named “Scott” is used and a Database “asdb”, the test entries are:

Username: *scott*

Password: *tiger*

Database: *asdb*

**Problem:** How do I create a user-specific RAD?

**Solution:** Complete the steps below to create a user-specific RAD:

### **To create a user-specific RAD**

1. Access the OIDDAS console, as usual.
2. Select “Directory” tab found at the top right hand corner of the page.
3. Click “Create” to create a new user.
4. Select a username lets say “ssotest” with a password of “ssotest1”.  
You can choose to add all other details.
5. Scroll down to the bottom of the page to “Resource Access Information”.
6. Click “Create” to create a new resource file .
7. Enter a Resource Name, for example, “ssotest\_db”.
8. Click Next and fill in the userid/password and connect string for the database, then click submit.

Userid over here is valid db user. For test purpose default Scott schema can be used. Database is the db used. Default is asdb. Thus the test entries could be:

Username: *scott*

Password: *tiger*

Database: **asdb**





# 6 Integrating Plumtree Corporate Portal

This chapter provides an overview of integrating COREid with Plumtree Corporate Portal 4.5WS.

This chapter covers the following topics:

- “About the Integration with Plumtree Portal” on page 349
- “Supported Versions and Platforms” on page 352
- “Enabling Single Sign-On in Plumtree 4.5” on page 352
- “Setting Up COREid to Protect Plumtree 4.5” on page 367
- “Enabling Single Sign-on in PlumTree 5.0.4” on page 370
- “Setting Up COREid to Protect Plumtree 5.0.4” on page 376
- “Integrating Other COREid Features” on page 378

## About the Integration with Plumtree Portal

The integration between COREid and the Plumtree Corporate Portal provides companies with a Web enterprise solution for building customized, secure business portals with integrated, identity-based Web access management.

In the integrated solution, the Plumtree Corporate Portal acts as a gateway to an enterprise intranet or extranet, providing users centralized access to a broad variety of applications and content hosted by the enterprise.

COREid provides a robust identity management and access security system to accurately track and manage the identities of Plumtree’s employees, customers, and partners. COREid also provides a common enterprise security and user identity infrastructure that controls access to the Plumtree Corporate Portal as well as to other enterprise applications and resources.

The integration supports single sign-on (SSO) between the applications within the portal framework and the enterprise Web applications that are secured by COREid.

The integration between COREid and Plumtree offers these major benefits:

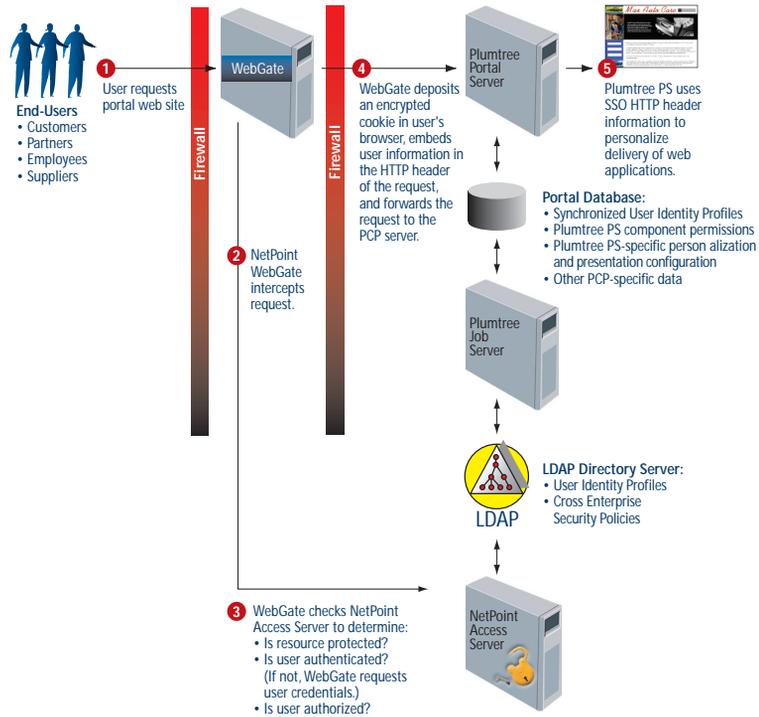
- **Single Sign-On**—COREid’s SSO services offer authorized users a secure connection with minimal authentication challenges to the resources that they need. Users need to log in only once to gain access to all resources that they need at a given level of authentication. This unhindered, seamless access provided by COREid’s SSO improves user efficiency, productivity, and user satisfaction.
- **Delegated Administration**—COREid offers delegated administration capabilities that complement those of the Plumtree portal. COREid’s delegated administration distributes responsibilities for managing identity and security policy information. Users can update some of their information; managers, suppliers, and partners can set up, change, and delete user identity information.
- **User and Group Management**—COREid manages an LDAP directory server that contains user identity information. The Plumtree portal synchronizes its own user database with the LDAP directory. This eliminates the need to create and manage separate identity profiles for each application.

COREid also provides strong group management capabilities, including support for static, dynamic, nested, and hybrid groups. COREid manages the group information that the Plumtree application uses to personalize the portal. Group affiliation is immediately reflected in the portal without manual changes.

- **Centralized Security Management**—COREid provides a common security management platform for all applications within an enterprise. This facilitates the maintenance of consistent security policies across an entire enterprise.
- **User Personalization**—COREid enables you to personalize portal content based on any user attribute, such as job title.

Integrating the Plumtree Corporate Portal with COREid does not change the users’ experience in Plumtree. Users can continue to access the portal guest pages without logging into Plumtree corporate portal. When a user attempts to log in, COREid intercepts the request and uses an authentication scheme to determine whether the user is authorized to access the portal. Those who are not authorized are denied access to Plumtree portal.

The diagram shown below illustrates how COREid components protect the Plumtree Corporate Portal and gadgets.



## Task overview: Integrating with Plumtree

1. Enabling SSO for the Plumtree portal as described in “Enabling Single Sign-On in Plumtree 4.5” on page 352.

In the Plumtree installation, you set up SSO and LDAP authentication sources, edit configuration files to support SSO, and then synchronize data from the COREid LDAP directory with the data in the Plumtree database.

2. Setting up COREid to protect the Plumtree portal as described in “Setting Up COREid to Protect Plumtree 4.5” on page 367.

In the COREid installation, you create policies that specify the content that you want to protect. Policies are created in the Access Manager.

3. (Optional.) Allow anonymous users to view the portal guest pages and creating a banner for the portal as described in “Integrating Other COREid Features” on page 378.
4. (Optional.) Personalize user pages, and embed COREid identity management functions as described in “Personalizing User Pages” on page 387.

# Supported Versions and Platforms

NetPoint 6.5 supports the Plumtree Corporate Portal 4.5WS on Windows 2000 SP4.

COREid 7.0.4 supports the PlumTree Corporate Portal 5.0.4WS on Windows 2000 SP4.

## Enabling Single Sign-On in Plumtree 4.5

Enabling SSO requires completion of several procedures on the Plumtree portal v 4.5.

### **Task overview: Enabling SSO for Plumtree 4.5**

1. “Creating an SSO Authentication Source” on page 352
2. “Creating an LDAP Authentication Source” on page 354
3. “Editing Configuration Files to Support SSO” on page 359
4. “Synchronizing LDAP Data with Plumtree Database” on page 363

## Creating an SSO Authentication Source

Authentication is the process of users proving their identity to a server. After users present their credentials to the server, authentication plug-ins process those credentials.

To enable COREid to authenticate users and groups on Plumtree, you must create an SSO authentication source so that COREid can authenticate users and groups in the Plumtree portal. To do this, you must first create an SSO password to use when you configure the SSO authentication source.

### **To create an SSO Password**

1. Open the Plumtree Administrator Control Panel application and click Start > Settings > Control Panel > Plumtree Administrator.

The Plumtree Administrator dialog box appears.

2. Click the Single Sign-On tab.
3. Enter an SSO secret key.

This secret key can be any string of characters. Make note of the string.

4. Click OK to close the Plumtree Administrator dialog box.

## To create an SSO authentication source on Plumtree

1. Log in to Plumtree as the administrator.  
Click Administration. The Administration Menu appears.
2. Click Authentication Sources > Add Single Sign-on Authentication Source.  
The Authentication Source Wizard appears.
3. On the General Info page, enter a name and description for the new authentication source.  
Describe the source carefully, as this description appears in a drop-down list during Authentication Source setup.

The screenshot shows the Plumtree Administration interface. At the top, there is a navigation bar with links for 'Portal Settings', 'Administration', 'Login As A Different User', and 'Logoff'. The date is 'Thursday, October 24, 2002'. Below this is a search bar and a 'Go' button. The main content area is titled 'ADMINISTRATION' and has a sub-header 'AUTHENTICATION SOURCES : EDIT SINGLE SIGN-ON AUTHENTICATION SOURCE : AUTHENTICATION SOURCE EDITOR'. On the left, there is an 'Editor Menu' with options: 'GENERAL INFO', 'OPTIONS', 'SYNCHRONIZATION SETTINGS', and 'SECURITY'. The main area is titled 'Authentication Source Editor' and shows the configuration for 'OblixSSOAuthnSource'. It includes a 'GENERAL INFO' section with a description: 'Enter a name and description for this Authentication Source. Users select an Authentication Source on the Login page based on this description.' Below this are two input fields: 'Authentication Source Name' and 'Authentication Source Description', both containing the text 'OblixSSOAuthnSource'. A note states: 'This Authentication Source uses the Single Sign-on Authentication Source Provider.' At the bottom, there is a link: 'Click here for help on the General Info page!' and three buttons: 'NEXT >>', 'FINISH', and 'CANCEL'.

4. Click Next. The Options page appears.
5. Enter the SSO password.
6. Click Validate Options to confirm that this password matches the secret key you entered in the Plumtree Administrator Control Panel.
7. Click Next until the Security Page appears.
8. Specify the Users and Groups who can see or edit this authentication source.  
Or, accept the defaults such as the Administrators Group and click Finish.

## Creating an LDAP Authentication Source

To update the Plumtree database with the current user and group information, you must synchronize users and groups in the Plumtree database with information in the LDAP directory. To do this, you must first create an LDAP Authentication Source to import users and groups data from the LDAP directory into your Plumtree portal.

### To create an LDAP authentication source on Plumtree

1. Log in to Plumtree as the administrator and click Administration.  
The Administration Menu appears.
2. Click Authentication Sources.  
Folders containing existing authentication sources appear. Click the folder to view its contents.
3. To create a new LDAP authentication source, click Add LDAP Authentication Source.  
The Authentication Source Wizard appears.
4. On the General Info page, enter a name and description for this authentication source.  
Word the description carefully, because this description appears in a drop-down list on the Login page of your portal. Users with accounts imported through this authentication source must select the same authentication source to log in to your portal.
5. Click Next. The Options page appears.

[GENERAL INFO](#)

[OPTIONS](#)

[DEFAULT PROFILE](#)

[SCHEDULE](#)

[SYNCHRONIZATION SETTINGS](#)

[FULLY SYNCHRONIZED GROUPS](#)

[SECURITY](#)

### OblixLDAPAuthnSource

**OPTIONS**  
Set the options for this Authentication Source. Required fields are marked with a \*.

Authentication Source Category:\*

Server Address:\*

Security Mode:\*

User Query Base:

User Query Filter:\*

User Name Attribute:\*

User Authentication Name Attribute:

Group Query Base:

Group Query Filter:\*

Group Name Attribute:\*

Group Membership Attribute:\*

Plumtree LDAP User's Authentication Name:

Plumtree LDAP User's Authentication Password:

LDAP Protocol Version:\*

Alternate LDAP Port:

SSL Certificate Database Path:

Normalize DN Mode:

Save this template as:

**6. On the Options page, enter values for your LDAP server as follows:**

Authentication Source Category	<p>The name of the users or groups that you want to import from this domain.</p> <p>For example, if you enter myDomain, all users imported from this source are placed in a users folder named myDomain, and all groups imported from this source are placed in a groups folder named myDomain.</p>
Server Address	<p><i>directory_server hostname</i> where <i>directory_server hostname</i> is the name or IP address of the directory server host.</p> <p>For example, myServer or 192.168.2.3.</p>
Security Mode	1 (for cleartext password authentication)

User Query Base	<p>The searchbase of the LDAP query that will return all of the users that you want to synchronize. The searchbase defines the starting point for searches in the User Manager application.</p> <p><i>ou=YourPeople, o=YourCompanyName</i>  where <i>ou=YourPeople, o=YourCompanyName</i> is your user searchbase.</p>
User Query Filter	<p>A filter that will limit a query result set to only the users you want to import.</p> <p>For example, <i>objectclass=YourPersonObjectClass</i>  where <i>YourPersonObjectClass</i> is your user object class.</p>
User Name Attribute	<p>The name of the attribute that contains the name of the user. This is LDAP name of the attribute assigned to the semantic type Login, for example, uid.</p>
User Authentication Name Attribute	<p>The name of the attribute used to bind to the LDAP directory. This is the name attribute in the PersonObject class, for example, cn.</p>
Group Query Base	<p>The searchbase of the LDAP query that will return all of the groups that you want to synchronize. The searchbase defines the starting point for searches in the Group Manager application.</p> <p><i>ou=YourGroups, o=YourCompanyName</i>  where <i>ou=YourGroups, o=YourCompanyName</i> is your group searchbase.</p>
Group Query Filter	<p>A filter that limits a query result set to only the groups you want to import.</p> <p><i>objectclass=GroupObjectClass</i>  where <i>GroupObjectClass</i> is your group object class, for example, GroupofUniqueNames.</p>
Group Name Attribute	<p>The LDAP attribute for the Group Object class, for example, cn.</p>
Group Membership Attribute	<p>The name of the attribute that contains a user's group membership information. Member or UniqueMember (the LDAP name of the Member or UniqueMember attribute.)</p> <p>For the active directory, use the Member name; for other directories, use the UniqueMember name.</p>
Plumtree LDAP User's Authentication Name	<p><i>Bind DN</i>  where <i>Bind DN</i> is the Root DN that you set in the System Configuration tab of the COREid System Console, for example, cn=Directory Manager.</p>

Plumtree LDAP User's Authentication Password	Root password of the Bind DN.
LDAP Protocol Version	3 (The LDAP version you are running.)
Alternate LDAP Port	The alternate port to which this LDAP directory connects. Default port is 389.
SSL Certificate Database Path	Leave this field blank.
Normalize DN Mode	Leave this field blank.
Save this template as	Leave this field blank.

- To confirm that the domain you entered is correct, click **Validate Options**.

The portal will try to find the domain, and then display a message stating whether it could connect successfully or not.

- If the validation is unsuccessful, click **Back** to check for errors in your settings.
- If the validation is successful, click **Next**.

The **Default Profile** page appears.

- On the **Default Profile** page, select a profile from the drop-down list.

This list is populated with the **Users** in the **Default Profiles Users** folder. Users imported through this authentication source will be given the specified profile.

- Click **Next**.

The **Schedule** page appears.

- On the **Schedule** page, click on **Create a Job for the Authentication Source**, and then click **Next**.

The **Synchronization Settings** page appears.

- Select **Synchronization Only**.

From the drop-down list for **Select an Authentication Partner**, select the SSO authentication source you created earlier.

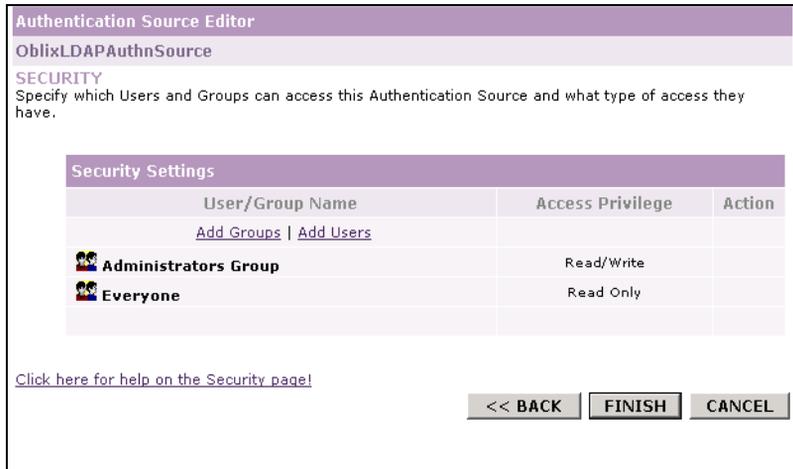
- Select **Full Synchronization** and click **Next**.

The **Fully Synchronized Groups** page appears.

13. Click Next.

The Security page appears.

14. Specify which Users and Groups can see or edit this authentication source or use the default groups.



**Authentication Source Editor**  
OblixLDAPAuthnSource

**SECURITY**  
Specify which Users and Groups can access this Authentication Source and what type of access they have.

**Security Settings**

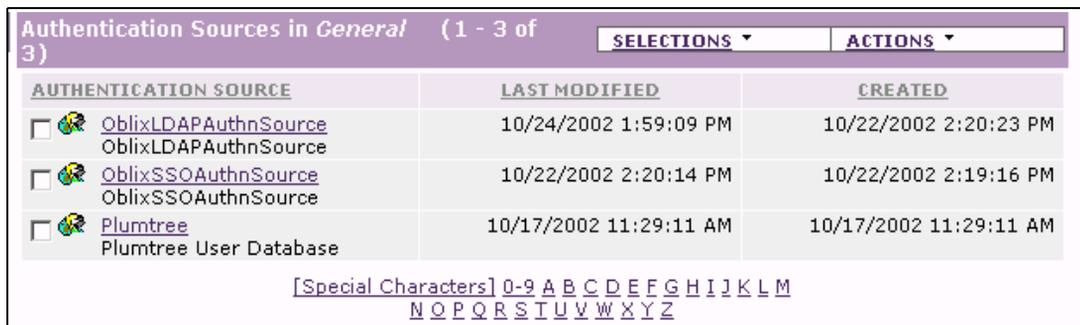
User/Group Name	Access Privilege	Action
<a href="#">Add Groups</a>   <a href="#">Add Users</a>		
 <b>Administrators Group</b>	Read/Write	
 <b>Everyone</b>	Read Only	

[Click here for help on the Security page!](#)

<< **BACK**   **FINISH**   **CANCEL**

15. Click Finish.

The Folders of Authentication Sources page appear. The authentication source that you created is listed in the folder in which it was created.



Authentication Sources in *General* (1 - 3 of 3)

AUTHENTICATION SOURCE	LAST MODIFIED	CREATED
<input type="checkbox"/>  <a href="#">OblixLDAPAuthnSource</a> OblixLDAPAuthnSource	10/24/2002 1:59:09 PM	10/22/2002 2:20:23 PM
<input type="checkbox"/>  <a href="#">OblixSSOAuthnSource</a> OblixSSOAuthnSource	10/22/2002 2:20:14 PM	10/22/2002 2:19:16 PM
<input type="checkbox"/>  <a href="#">Plumtree</a> Plumtree User Database	10/17/2002 11:29:11 AM	10/17/2002 11:29:11 AM

[Special Characters] 0-9 A B C D E F G H I J K L M  
N O P Q R S T U V W X Y Z

By default, the job is unscheduled. You must decide how and when you want the job to run as described in “Synchronizing LDAP Data with Plumtree Database” on page 363.

## Editing Configuration Files to Support SSO

After you have created an SSO authentication source, you must direct Plumtree to use it. You do this by editing values in certain configuration files in your Plumtree installation described below.

---

**Note:** You must restart the portal Web server after you create or edit configuration files to enable the changes to take effect.

---

- **config.xml**— Directs Plumtree to log on using single sign-on. The config.xml file is located the directory *plumtree\_home*\4.5WS\PortalPages, where *plumtree\_home* is the location of your Plumtree installation.

Edit the values for the single sign-on parameters as shown below. You must do this for each virtual directory that you want to enable for single sign-on.

Parameter	Value
INTSSOVENDOR	3. This is the identity of COREid.
STRDEFAULTAUTHSOURCEPREFIX	Enter the value of the authentication source category that you used for the LDAP synchronizing source; for example, LDAPUsers.
STRSSOCOOKIEDOMAIN	Enter the value that you set for the primary HTTP cookie domain for the WebGate instance that protects the Plumtree portal; for example, .example.com  You set the domain in the Access System Console. See the <i>Oracle COREid Administration Guide Volume 2</i> for more information.

- **oblix.asp**— extracts the header variable that you specify in the oblix.asp file and passes it on to Plumtree to enable SSO. Oblix.asp is located in *plumtree\_home*\4.5WS\PortalPages\PortalPages\ss\vendors.

where *plumtree\_home* is the location of your Plumtree installation

In the following example, HTTP\_OBLIX\_UID is the header variable that has been configured to be returned as a header variable in the policy domain that is protecting the portal. See “To create a policy domain” on page 368 for more information.

Original code segment	Modified code segment
<pre>Function GetOblixUserName GetOblixUserName = ExtractFromHeaders("UID") End Function</pre>	<pre>Function GetOblixUserName dim somevariable somevariable = ExtractFromHeaders("HTTP_OBLIX_UID") GetOblixUserName=trim(somevariable) End Function</pre>

**Warning:** You may get a login error if an extra character is appended to the end of the header variable (for example, User WWM Data\reed&). In such a case, instead of the modifying the code as documented above, modify the code as shown below.

```
Function GetOblixUserNam
Dim myLen
myLen = Len(ExtractFromHeaders("UID"))
myLen = CInt(MyLen) - 1
GetOblixUserName = Left(ExtractFromHeaders("UID"), myLen)
End Function
```

and set the value of Dim arrSecureHeaders ( ) to 0 [zero] as shown below:

```
Function GetOblixSecureHeadersArray
Dim arrSecureHeaders(0)
GetOblixSecureHeadersArray = arrSecureHeaders
End Function
```

You must create or modify the following configuration files to integrate the COREid SSO feature in your portal to redirect anonymous users as guests to your portal page.

**Note:** It is recommended that you backup configuration files before you modify them.

### To create loginoblix.asp

1. On the portal server, create a text file named `loginoblix.asp` located in `plumtree_home\4.5WS\PortalPages\PortalPages\admin` where `plumtree_home` is the path to your Plumtree installation.
2. Add the following code to the file:

```
<%
Session("blnSSOLoginFailed") = False
response.redirect("../sso/sso.asp")
%>
```

## To edit dologin.asp

1. Open the file dologin.asp located in  
*plumtree\_home*\4.5WS\PortalPages\PortalPages\admin  
where *plumtree\_home* is the path to your Plumtree installation.
2. Insert the code shown in bold before the last line (End If):

```
...  
If (Session("bAttemptingSSOLogin") = True) Then  
    Session("bAttemptingSSOLogin") = False  
    If Catch(&h80044401) Then  
        ' this means the SSO account has not been synced  
  
    Response.Redirect("login.asp?UserID="+strUserID+"&RedirectURL=" +  
    Application("AppUtil").StringEncode  
    Server,strRedirectURL,c_intURLEncode) + "&Mode=8")  
        End If  
        If Catch(&h80044402) Then  
            ' this means SSO is set up incorrectly -- if the password  
            was wrong, that means the sso password is wrong  
  
    Response.Redirect("login.asp?UserID="+strUserID+"&RedirectURL=" +  
    Application("AppUtil").StringEncode(Server,strRedirectURL,c_int  
    URLEncode) + "&Mode=3")  
        End If  
  
    If Catch(&h80044405) then  
        ' this SSO user has been locked, meaning it is the 'guest'  
        SSO user so redirect to the my page as our guest  
        Response.Redirect("../mypage/mypage.asp?UserID=2")  
        End If  
    End If
```

## To create dologout.asp

1. Copy the file dologoff.asp located in  
*plumtree\_home*\4.5WS\PortalPages\PortalPages\admin  
where *plumtree\_home* is the path to your Plumtree installation.
2. Save the copy as dologout.asp in  
*plumtree\_home*\4.5WS\PortalPages\PortalPages\admin



```

        &RedirectURL=<%=g_objAppUtil.StringEncode(Server,
Request.ServerVariables("URL") & "?" &
Request.ServerVariables("QUERY_STRING"), c_intURLEncode)%> "
        TARGET=_top><!-- Login
--><%=g_objSessionIntlUtil.GetString(22, PTICOMMON)%></A>
        <else%>
        <A HREF="<%=g_strApplicationBaseURL & "admin/
loginoblix.asp?UserID=" &
strUserID%>&RedirectURL=<%=g_objAppUtil.StringEncode(Server,Req
uest.ServerVariables("URL"),c_intURLEncode)%>" TARGET=_top><!--
Login --><%=g_objSessionIntlUtil.GetString(22, PTICOMMON)%></A>
        <%End If
        Else %>
        <A HREF="<%=g_strApplicationBaseURL & "admin/
loginoblix.asp?UserID=" & strUserID%>"><!-- Login
--><%=g_objSessionIntlUtil.GetString(22, PTICOMMON)%></A>
        <%End If
        ...

```

## Synchronizing LDAP Data with Plumtree Database

Data synchronization is important because updated information on group memberships and users is crucial for portal access and personalization.

You can synchronize data manually or automatically at specific time intervals. To synchronize manually, you schedule a job and run it once for immediate synchronization. For periodic synchronization, you schedule a job to run at specified time intervals.

After you run a job, you can view its status to see if it ran successfully. When a job has run successfully, you will be able to view the replicated LDAP data in the Plumtree database.

---

**Caution:** Initial full synchronization can take a long time if you have many entries. You must fully synchronize at least once to enable SSO between Plumtree and COREid.

---

### To automatically synchronize data

1. On the Plumtree portal machine, go to Control Panel > Services, and ensure that the Plumtree Job Dispatcher service has started.
2. Log in to Plumtree as the administrator and click Administration.
3. In the Administration Menu, click Jobs > Add Job.

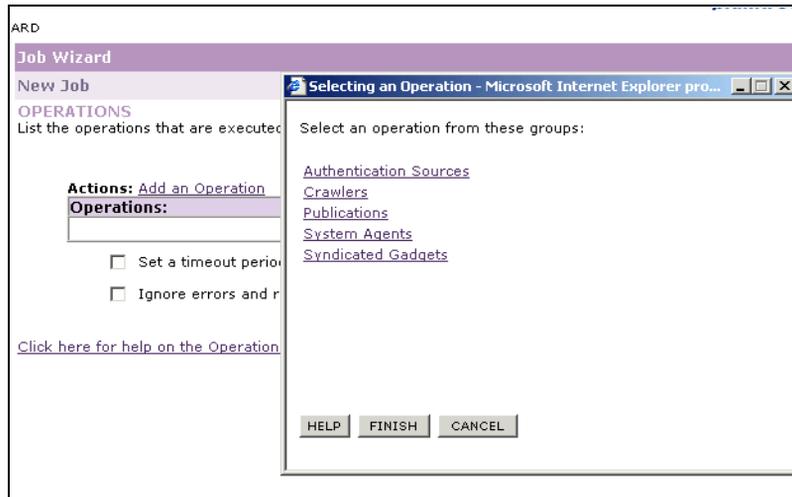
The General Info page appears.

4. Enter a name and description for the new job.
5. Click Next.

The Operations page appears.

6. Click Add an Operation.

The Selecting an Operation pop-up window appears.



7. Click Authentication Sources.  
The existing authentication sources appear.
8. Select the LDAP authentication source that you created and click Finish.  
The Operations page reappears.
9. Click Next.  
The Schedule page appears.
10. Specify the time period when the synchronization job is to be run.

**Job Editor**  
**OblixLDAPAuthnSource Job created on...**

**SCHEDULE**  
 Schedule this Job to run.

**Unscheduled**

---

**Run Once**      Date:       Time:

---

**Run Periodically**

**Next Run:**      Date:       Time:

**Run Every:**     

**Don't Run After:**      Date:       Time:

**Other Restrictions:**     

11. Click Next until the Security page appears.
12. Click Finish.
13. If you click Jobs in the Administration Menu, the General folder appears on the right.

Your synchronization job is listed in the folder. Click the folder to see its contents.

### To manually synchronize data

1. In the General folder, checkmark the check box next to your synchronization job, and put your cursor over Actions.

A drop-down list of the actions appear.

2. Select Run Once; your job will be scheduled to run once, immediately.

It will also show you a new job created called:

Run once: Copy of *your job name*

where *your job name* is the name of your job.

When the job is done, the job copy is removed from the list of jobs, indicating that it has been completed.

## Viewing Synchronized Information

The following tasks describe how to view the status of the synchronization job and the updated Plumtree database.

### To view the status of a job

1. Click Jobs and then click Job Server Manager.

The Job Server Manager View appears.

2. Click Job Histories View link.

A list of all jobs (past and present) appears. If a job is still in progress, its status will show as running.

After synchronization has been completed, you should be able to see all the users and groups from the LDAP data source replicated in the Plumtree database.

### To view the updated Plumtree database

1. In the portal, click Administration.
2. In the Administration Menu, click Users or Groups.

A folder with the name of the source that you had given earlier is displayed; for example, LDAP Users.

3. Click the folder to view the list of users or groups.

The names will be prefixed with the LDAP source name; for example, LDAPUsers\Accounting Managers.

In a Group folder, you can also view the members of the groups.

Users in LDAPUsers (1 - 200 of 725)		SELECTIONS ▾	ACTIONS ▾
USER		LAST MODIFIED	CREATED
<input type="checkbox"/>  <a href="#">LDAPUsers\Admin</a> CN=MASTER ADMIN,O=COMPANY,C=US		9/24/2002 5:05:56 PM	9/24/2002 5:03:13 PM
<input type="checkbox"/>  <a href="#">LDAPUsers\anonymous</a> CN=ANONYMOUS,O=COMPANY,C=US		9/24/2002 5:03:13 PM	9/24/2002 5:03:13 PM
<input type="checkbox"/>  <a href="#">LDAPUsers\atop1</a> CN=ATOP1,O=COMPANY,C=US		9/27/2002 11:48:37 AM	9/27/2002 11:48:37 AM
<input type="checkbox"/>  <a href="#">LDAPUsers\markh</a> CN=MARK,O=COMPANY,C=US		9/27/2002 4:19:18 PM	9/27/2002 4:19:18 PM

# Setting Up COREid to Protect Plumtree 4.5

Generally in the Plumtree Corporate Portal, users click the Login button to log in to the portal. Once they are authenticated, they can view their personalized pages. To log out of the portal, users click the Logout button.

In another configuration, you might want all users to see a *guest* portal and authenticate users only when they log into the portal. If you have enabled user access to the guest pages, then any user can go the main portal page and view those guest pages without logging into the portal. For more information, see “Enabling Anonymous Users to View Portal Guest Pages” on page 385.

When a user attempts to login, the COREid authentication policy challenges the user. Once COREid authenticates the user, it checks to see if the user is authorized. If the user is authorized, an ObSSOCookie and a header variable are sent to enable SSO into Plumtree. The user is then logged in to Plumtree.

You can configure the Plumtree Corporate Portal so that when users log out, they are automatically logged out of both Plumtree and COREid.

To setup COREid’s SSO service for Plumtree, after you have installed COREid, you must create policies in the Access Manager that specify the content you want to protect.

Setting up COREid to protect Plumtree consists of the following tasks:

- “Installing COREid Components” on page 367
- “Creating a Policy Domain” on page 368

## Installing COREid Components

In order to integrate COREid with Plumtree, you must install the following COREid applications:

- COREid Server
- Access Server
- WebPass
- Access Manager
- WebGate

To enable COREid to protect the portal, install a WebGate on the Plumtree Corporate Portal Web server.

---

**Note:** You can install COREid, WebPass, and Access System on the same server. However, it is recommended that you do not install the WebGate and WebPass on the server where Plumtree has been installed.

---

For more information on installing and configuring COREid, see the *Oracle COREid Installation Guide*.

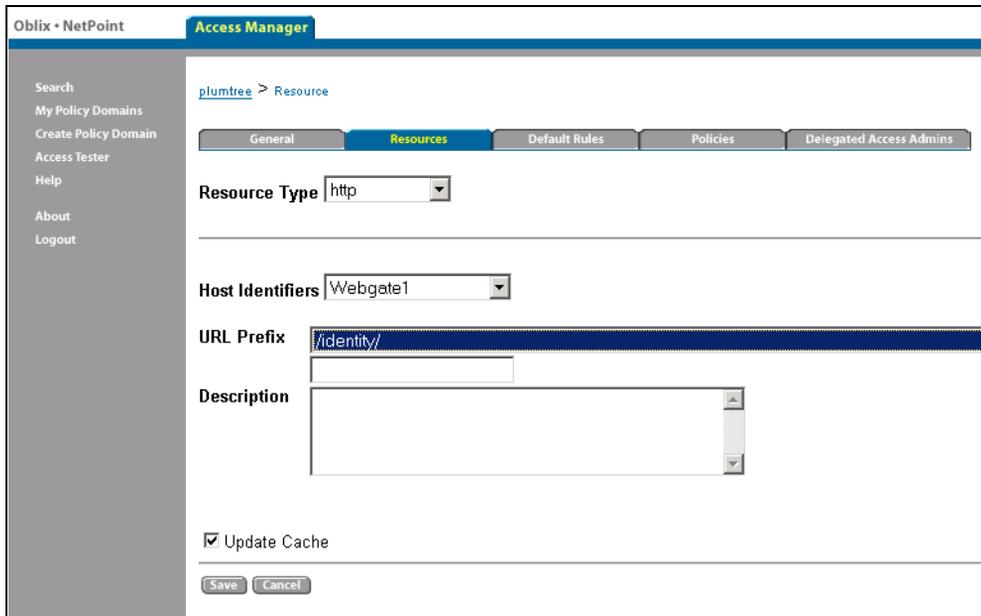
## Creating a Policy Domain

To protect content in the Plumtree portal, you must set up a policy domain in COREid. A policy domain encompasses a logical set of content that you want to protect, the rules for authentication and authorization protection, the policies for protection, and the administrative rights.

See the *Oracle COREid Administration Guide Volume 2* for more information on creating policy domains.

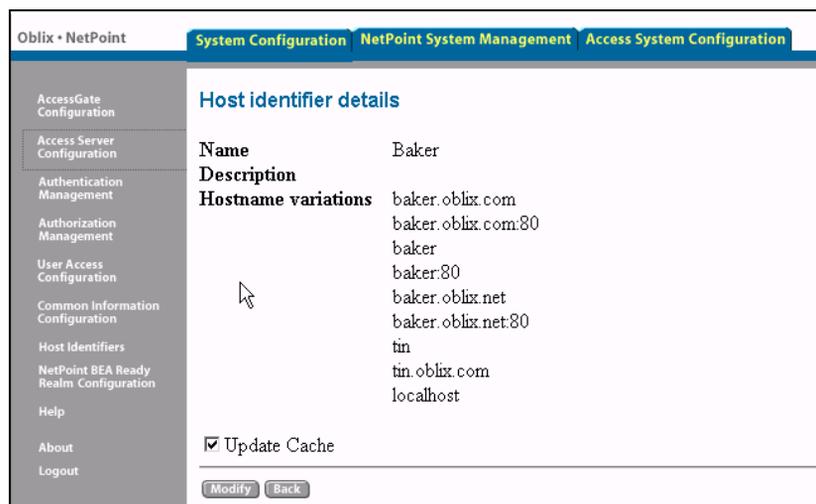
### To create a policy domain

1. Launch the Access Manager and click Create Policy Domain.
2. In the General tab, enter a name and description for the new policy domain.
3. In the Resources tab, select HTTP as the resource type.



4. To use a host identifier, create a host identifier in the Access System Console > Access System Configuration > Host Identifiers.

You must add a fully qualified host name as one of the host name variations; for example, plum1.oblix.com. See the *Oracle COREid Administration Guide Volume 2* for more information.



5. Select a URL prefix from the drop-down list. This will be the entry point to the Plumtree Corporate Portal. For example, /portal or /portal45.

---

**Note:** Ensure that the gadget URL and policy URL are identical.

---

6. In the Default Rules tab, click Authentication Rule and select a rule from the Authentication Scheme drop-down list.

Select an authentication scheme such as Basic over LDAP.

7. In the Default Rules tab, select Authorization Rules and add a new rule that allows access to anyone.

For more information on creating authorization rules, see the *Oracle COREid Administration Guide Volume 2*.

8. In the Authorization Rules tab, select Actions, and add a header variable to facilitate SSO.

You must specify this header variable in the file oblix.asp (described in “Editing Configuration Files to Support SSO” on page 359) to direct Plumtree to look for this header variable. For example, you can add a header variable called UID that returns the loginid attribute (the LDAP name of the attribute with a semantic type of Login).

9. From Access Manager > My Policy Domain, click the check box for the domain to be enabled and then click Enable.



The screenshot shows the Oracle Access Manager configuration interface. The 'Default Rules' tab is selected, and the 'Authorization Rules' sub-tab is active. Below the navigation tabs, there is a section for 'Actions' with a 'Custom Actions' link. Underneath, the 'Authorization Success' section contains a table with the following data:

Return	Type	Name	Return Attribute
	headerVar	HTTP_OBLIX_UID	genuserid
	headerVar	FULL_NAME	cn

At the bottom of the configuration area, there is a checkbox labeled 'Update Cache' which is checked.

## Enabling Single Sign-on in PlumTree 5.0.4

You enable SSO on the Plumtree portal by completing the following tasks:

- “Creating an SSO Authentication Source” on page 371
- “Creating an LDAP Authentication Source” on page 372
- “Editing Configuration Files to Support SSO” on page 373
- “Synchronizing LDAP Data with Plumtree Database” on page 374

- “Enabling SSO Logout” on page 375

## Creating an SSO Authentication Source

Authentication is the process of users proving their identity to a server. After users present their credentials to the server, authentication plug-ins process those credentials. To enable COREid to authenticate users and groups on Plumtree, you must create an SSO authentication source so that COREid can authenticate users and groups in the Plumtree portal. To do this, you must first create an SSO password to use when you configure the SSO authentication source.

### To create an SSO Password

1. Launch the Plumtree Administrator Control Panel application and navigate to Start > Settings > Control Panel > Plumtree Administrator.

The Plumtree Administrator dialog box appears.

2. Click Single Sign-On.
3. Enter an SSO secret key.

---

**Note:** This secret key can be any string of characters. Make note of the string.

---

4. Click OK to close the Plumtree Administrator dialog box.

### To create an SSO authentication source on Plumtree

1. Log in to Plumtree as the administrator. Click Administration. The Administration Menu appears.
2. Click Authentication Folder. In the Create Object drop down box select Authentication Source-SSO. The Authentication Source Wizard appears.
3. Enter the SSO password.
4. Click Validate Options to confirm that this password matches the secret key you entered in the Plumtree Administrator Control Panel.
5. On the Properties and Names page, enter a name and description for the new authentication source. Describe the source carefully, as this description appears in a drop-down list during Authentication Source setup.
6. Specify the properties for the Authentication Source.

## Creating an LDAP Authentication Source

To update the Plumtree database with the current user and group information, you must synchronize users and groups in the Plumtree database with information in the LDAP directory. To do this, you must first create an LDAP Authentication Source to import users and groups data from the LDAP directory into your Plumtree portal. To create an LDAP authentication source on Plumtree.

1. Log in to Plumtree as the administrator and click Administration.

The Administration Menu appears.

2. Click Authentication Folder. In the Create Object drop down box, select Authentication Source-SSO. The Authentication Source Wizard appears.
3. In the Authentication Source Category box, type the prefix used to distinguish the users and groups imported from this domain. For example, if you enter myDomain, each user name and each group name will be prefixed by the string myDomain. Thus, myUser becomes myDomain\myUser and myGroup becomes myDomain\myGroup.

You can set the category to any value you want, but after you create this authentication source, you cannot change this value.

4. Templates can populate configuration options with common default values. To use an existing template to populate the values in this editor, choose one from the Template drop-down list.
5. Add values under LDAP Settings to facilitate portal access to the LDAP server from which you are importing users. Consult online help for instructions.
6. To confirm the domain you entered, click Validate Options.

After the portal has attempted to find the domain, it displays a message stating whether it connected successfully or not.

- If the validation fails, check for errors in your settings.
  - If the validation succeeds, go to the Synchronization page.
7. Select the SSO authentication source you created earlier from the drop-down list labelled Select an Authentication Partner. Select Full Synchronization.
  8. On the Schedule page, select Create a Job/Add a Job for this Authentication Source. Consult online help for instruction on creating or adding a job.
  9. Add appropriate properties on the Properties and Names page.

## Editing Configuration Files to Support SSO

By default, the portal expects the COREid server to forward the user name header named uid. If you configure your COREid server to forward a user name header with a different name, you must configure your SSO implementation as BasicSSO service. For details about BasicSSO service, see the PlumTree Administration Guide.

Configure elements under the <Authentication> parent element in the PTConfig.xml, as detailed in Table 16

**Table 16** Parameters In Ptconfig.xml

Parameter	Value
SSOVendor	For COREid the SSOVendor value is 3.
DefaultAuthSourcePrefix	Use the same value you specified for Authentication Source Category when you configured your authentication source.
CookiePath	“/” (Specify a different setting only if your SSO authentication server requires a different convention.)
CookieDomain	Specify the fully qualified domain name to which you want the cookie forwarded. For example, if you specify .company.com, the cookie enables access to all domains that end in .company.com. If you specify .sub.company.com, the cookie enables access only to domains that end in .sub.company.com. The string must start with a period (.) and include a minimum of two periods.
SSOCookieIsSecure	0 or 1. 0 (the default) specifies the connection to the remote server does not require SSL for the cookie to be forwarded. 1 specifies SSL is required.

The following example enables integration with a COREid authentication server:

```
<SSOVendor value="3"/>
<DefaultAuthSourcePrefix value="HQ"/>
<CookiePath value="/" />
<CookieDomain value=".company.com"/>
<SSOCookieIsSecure value="0"/>
```

## Synchronizing LDAP Data with Plumtree Database

Data synchronization ensures the updating of information on group memberships and users, which is crucial for portal access and personalization. You can synchronize data manually or automatically at specific time intervals. To synchronize manually, you schedule a job and run it once for immediate synchronization. For periodic synchronization, you schedule a job to run at specified time intervals.

After you run a job, you can view its status to see if it ran successfully. When a job has run successfully, you can view the replicated LDAP data in the Plumtree database.

---

**Caution:** Initial full synchronization requires a long time if you have many entries.

---

You must fully synchronize at least once to enable SSO between Plumtree and COREid.

### To automatically synchronize data

1. On the Plumtree portal host, navigate to Control Panel > Services, and verify that the Plumtree Job Dispatcher service has started.
2. Log in to Plumtree as the administrator, then click Administration.
3. In the Administration Menu, navigate to the Administration folder > Jobs > Select the job you earlier created / added for the LDAP Authentication Source. The main settings page appears.
4. Specify the time period when the synchronization job is to be run.
5. The Edit Job User page also provides options for checking the job log and history through the Job History tab. It also lets you check security-related options.

### To manually synchronize data

1. On the Plumtree portal host, navigate to Control Panel > Services, and verify that the Plumtree Job Dispatcher service has started.
2. Log in to Plumtree as the administrator, then click Administration.
3. In the Administration Menu, click on the Administration folder > Jobs > Select the job you earlier created / added for the LDAP Authentication Source. The main settings page appears.
4. Select Run Once - Now option in the settings. This job runs once, starting immediately.

## Viewing Synchronized Information

After synchronization has been completed, you should be able to see all the users and groups from the LDAP data source replicated in the Plumtree database.

### To view the updated Plumtree database

1. In the portal, click Administration.
2. In the Administration Menu, click your administration folder.
3. Expand the Users tab. The user names will be prefixed with the LDAP source name.

#### For example:

```
LDAPUsers\Accounting Managers
```

In a Group folder, you can also view the members of the groups.

## Enabling SSO Logout

When a user clicks the PlumTree “Log Off” button in a COREid-protected PlumTree session, users are logged out from PlumTree, but the ObSSOCookie generated for the users is not killed. Hence, the COREid session for that particular user remains active. Users need to customize the PlumTree logout functionality to facilitate logout from PlumTree as well as COREid.

Consult “Customizing the Portal UI: Using Plumtree Event Interfaces (PEIs)” section in “Enterprise Web Development Documentation” on the PlumTree Website for information on modifying the PlumTree UI. Users need to implement the “OnBeforeLogout” interface as:

```
public virtual Redirect OnBeforeLogout(Object _oUserSession, ApplicationData
    _appData)
{
    PTDebug.Trace(Component.Portal_UI_Infrastructure, TraceType.Error,
        "Before logout event");
    String myCookie = _appData.GetCookie("ObSSOCookie");

    if(myCookie != null)
    {
        Redirect myredirect = new Redirect();
        myredirect.SetLinkToExternalURL("http://
            <NetPoint_Server_Name:port>/access/oblix/lang/en-us/
            logout.html");
        return myredirect;
    }
    return null;
}
```

# Setting Up COREid to Protect Plumtree 5.0.4

Typically, users click the Login button to log in to the Plumtree Corporate Portal. Once they are authenticated, they can view their personalized pages. To log out of the portal, users click the Logout button. In an alternate configuration, you might want all users to see a guest portal, then authenticate users only when they log into the portal. (If you have enabled user access to the guest pages, then any user can go the main portal page and view those guest pages without ever logging into the portal).

When a user attempts to login, the COREid authentication policy challenges the user. Once COREid authenticates the user, it checks to see if the user is authorized. If the user is authorized, an ObSSOCookie and a header variable are sent to enable SSO into Plumtree. The user is then logged in to Plumtree.

You can configure the Plumtree Corporate Portal so that when users log out, they are automatically logged out of both Plumtree and COREid. To setup COREid's SSO service for Plumtree, after you have installed COREid, you must create policies in the Access Manager that specify the content you want to protect.

Setting up COREid to protect Plumtree consists of the following tasks:

- "Installing COREid Components" on page 376
- "Creating a Policy Domain"

## Installing COREid Components

To integrate COREid with Plumtree, you must install the following COREid applications:

- COREid Server
- Access Server
- WebPass
- Access Manager
- WebGate

To enable COREid to protect the portal, install a WebGate on the Plumtree Corporate Portal Web server.

---

**Note:** You can install COREid server, WebPass, and Access System on the same server. However, Oracle recommends that you do not install the COREid server and WebPass on the server where Plumtree has been installed. For more information on installing and configuring COREid, see the COREid 7.0 Installation Guide.

---

## Creating a Policy Domain

To protect content in the Plumtree portal, you must set up a policy domain in COREid. A policy domain encompasses the following elements:

- a logical set of content to protect
- authentication and authorization protection
- policies for protection
- administrative rights.

See the COREid 7.0 Administration Guide for more information on creating policy domains.

1. Install the Oracle COREid 7.0 suite, which includes the following components: COREid, WebPass, Access Server, and Access Manager.
2. Launch COREid Access Manager, typically `http://oblix_access_server:port/access/oblix`.
3. Create the following objects in the order they are presented. Consult the *Oracle COREid Administration Guide* for details.

## Configuring the COREid WebGate

Configure your COREid WebGate following the procedure appropriate for your portal deployment in the Oracle *COREid Installation Guide*:

- WebGate for Apache
- WebGate for IIS

Use the version of COREid WebGate that is compatible with your COREid suite. For example, if you use Oblix NetPoint 6.5, configure Oblix WebGate 6.5; if you use COREid 7.0, use COREid WebGate 7.0.

### To set up the COREid WebGate for Apache:

1. On the host computer for the Portal Server, install the version of Apache required by the WebGate:
  - For WebGate 6.5, install Apache 1.3.
  - For WebGate 7.0, install Apache 1.3 or Apache 2.0.

---

**Note:** Note: the version of Apache provided by Plumtree and described in the Installation Guide for Plumtree Corporate Portal cannot be used with the COREid WebGate. You must download the required Apache version from the Apache Web site.

---

2. On the host computer for the Portal Server, install the COREid WebGate for Apache. For details, see the *Oracle COREid Installation Guide*.
3. On the Web application server to which the portal application is deployed, modify the Web application server setting to turn off URL rewrites. For details, refer to your Web application server documentation or Plumtree Knowledge Base article DA\_239501, "Configuring Web Application Servers to not Rewrite URLs."

## Configuring COREid WebGate for IIS

Install the version of the COREid WebGate that is compatible with your COREid suite. For example, if you use Oblix NetPoint 6.5, configure Oblix ISAPI WebGate 6.5; if you use COREid 7.0, use COREid ISAPI WebGate 7.0.

To set up WebGate for IIS, run the COREid WebGate for IIS installer on the host computer for the Portal Server.

## Integrating Other COREid Features

COREid offers several other features that you can integrate with Plumtree such as allowing guest users to view portal pages, personalizing user pages, and embedding other COREid identity management functions into your portal. You can also setup SSO to other portals, and manage passwords and self-registration.

The following tasks are discussed in this section:

- “Using Gadgets to Create Integration Features” on page 378
- “Enabling Anonymous Users to View Portal Guest Pages” on page 385
- “Personalizing User Pages” on page 387
- “Embedding COREid Management Functions” on page 389
- “Importing COREid Dynamic Groups” on page 393
- “Password Management” on page 396
- “Self-Registration” on page 396

## Using Gadgets to Create Integration Features

You use gadgets to personalize user pages. For example, you can create a gadget to greet a user when the individual logs into the Plumtree portal. You also use gadgets to create a self-registration interface where users can register themselves into the Plumtree portal or a lost password interface to enable users to reset their password if they lose it.

The file SampleGadgets.zip contains folders for various types of gadget files shown below.

---

**Note:** To access SampleGadgets.zip, go to the COREid customers site at <http://www.oracle.com/support/contact.html>.

---

Folder	File Name	Notes
Admin	dologin.asp dologout.asp loginoblix.asp	These files are used to integrate the SSO feature into your portal.
Common	banner.asp	
sso \ vendors	oblix.asp	
HelloUserGadget	gadget.asp	This file personalizes user pages.
iFramework	adminprefs.asp gadget.asp	These files create iFrame gadgets.
sso	userinfo.asp	
ChangePassword	adminprefs.asp gadget.asp	These files enable users to change their passwords.
CreateAccount	adminprefs.asp gadget.asp	These files are used for the self-registration feature.
LostPassword	adminprefs.asp gadget.asp	These files are used for the lost-password management feature.

## Creating Gadgets

You create a gadget in the Plumtree Administration Menu.

### To manually create a gadget

1. In the Administration Menu, click Gadgets.
2. Click Add Remote Gadget.  
The General Info page appears.
3. Enter a name and description for the new gadget.
4. From the Gadget Server drop-down list, select the server where the gadget is located.

Ensure that the enabled button is selected.



5. Click Next.

The HTTP Configuration page appears.

6. In the remote gadget field, enter URL for the gadget, (example, demo/gadget.asp).
7. Click Next through the next few pages until the Security page appears.
8. In the Security page, specify the users and groups to whom you want to grant read/write privileges and click Finish.

## Importing Gadgets

You can import gadgets from another portal server. Oblix provides sample gadgets that you can import. These gadgets are designed for ID management functions such as self-registration, changing passwords, creating groups, and workflow search.

These samples are located in the file oblixgadgets.zip. You can access oblixgadgets.zip at <http://www.oracle.com/support/contact.html>.

### Task overview: Before you import gadgets

1. In the Organization Manager, create a self-registration workflow and note the workflow ID.  
The gadget that you import uses this workflow to create user accounts. See the *Oracle COREid Administration Guide Volume 1* for information on creating workflows.
2. Unzip the file OblixGadgets.zip that contains sample gadgets and save the samples to a directory on your machine.
3. Download Plumtree ASP GDK3.5 from the Plumtree Web site (<http://workshop.plumtree.com>) and install it on the machine where you installed the gadgets.

4. On IIS, create a virtual directory on your machine such as OblixGadgets.  
Point the virtual directory to the actual directory where your sample gadgets from OblixGadgets.zip are located.

5. Open the OblixPlumtreeGadgets.xml file.

This file is located in the same directory as the sample gadgets.

6. In the XML file, in the line shown below, replace the server name with your server name:

```
"http%u003A%u002F%u002Fplum%u002D3%u002Eoblix%u002Ecom"
```

(unencoded, it stands for <http://plum-3.oblix.com>)

In the URL, the encoded values for non-alphanumeric characters are as follows:

- %u003A = :
- %u002F = /
- %u002E = .
- %u002D = -

### **To import gadgets**

1. Log in to the Plumtree Corporate Portal as the administrator.
2. Click Administration > Gadgets > Import Remote Gadget Package.

The General Info page appears.

3. In the URL for Gadget Package field, enter the following URL:

```
http://server/oblixgadgets/OblixPlumtreeGadgets.xml
```

This is the URL to the file that contains the gadget package.

4. Click Next.

The Choose Gadgets page appears.

5. Select the gadgets that you want to import from the gadget package.

6. Click Next.

The Review Package page appears. You can review the gadgets that you selected.

7. Click Next.

The Gadget Server Settings page appears.

8. Enter the gadget server base URLs as follows:

- For a gadget server with the UUID uri://oblix.com/2002/plum-2:  
http://server2/identity/oblix/
  - For a gadget server with the UUID uri://oblix.com/2002/plum-3  
http://server/oblixgadgets/
9. Click Finish to import the gadgets.

The following four gadget server containers are created:

- Oblix Admin Gadgets
- Oblix Gadgets (iFrame)
- Oblix Gadgets (without Gateway)
- Oblix Gadgets (with Gateway)

## Creating User Accounts

Before you can use the imported gadgets, you must specify the self-registration workflow to enable the creation of user accounts.

### To create user accounts

1. In the Administration Menu, click Gadgets > Oblix Admin Gadgets.
2. Open the Oblix Create Account gadget.
3. In the Alignment and Administrative Preferences page, click the Edit Administrative Preferences link.
4. Enter the workflow ID of the self-registration workflow that you created.
5. Review the COREid Identity Server Base URL, correct it if necessary, and click Finish.
6. Navigate back to the folder of gadgets and open the Oblix Gadgets folder (iFrame).
7. In the Alignment and Administrative Preferences page, click the Edit Administrative Preferences link.
8. In the iFrame URL, ensure that DN for the Create Group workflow is correct. If necessary, edit the HTTP settings to provide the correct DN.

You can now add these gadgets to My Pages and use them.

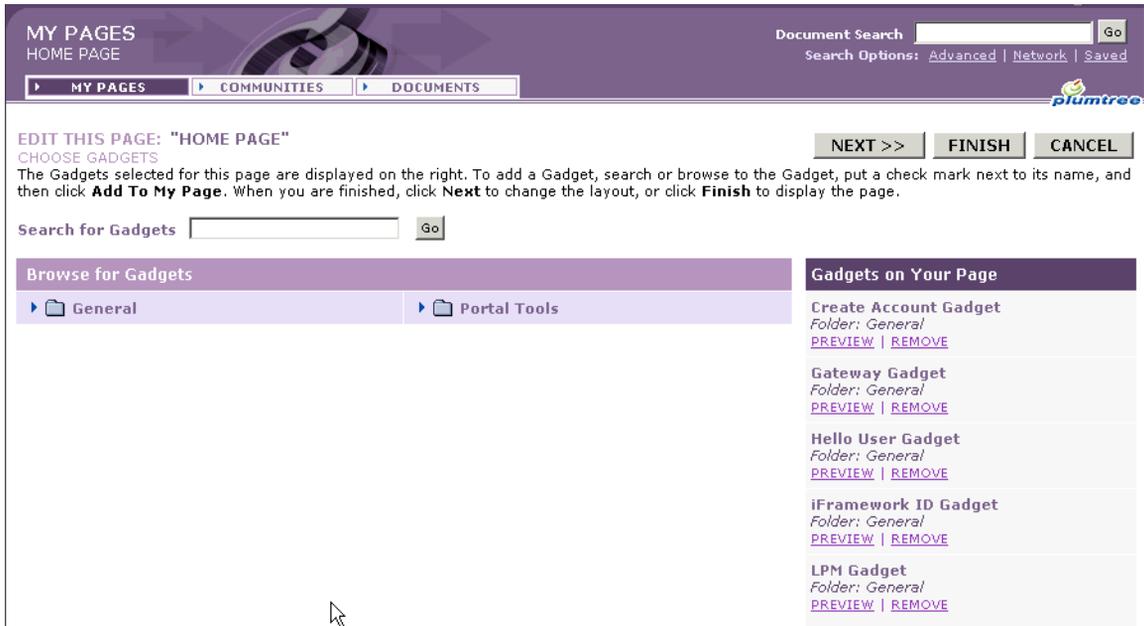
# Displaying Gadgets

After you have created a gadget, you can display the gadget on your page.

## To display a gadget

1. Click My Pages.
2. Put your mouse on My Pages again, select Edit this Page, and then select Choose gadgets.

The Choose Gadgets page appears.



3. Select the gadget that you want and click Finish.

This takes you to My Pages where you can view the gadget on the page.

---

**Note:** If there is a WebGate protecting the Web server where this Hello User gadget is located, then make sure you have set it up for SSO with the correct policy and header variables.

---

## Adding Gadgets to the Plumtree Portal Guest Pages

You can add gadgets to enable users to self-register themselves on the portal or to reset passwords.

### To add a gadget to the portal guest pages

1. Log in to the Plumtree portal as the administrator.
2. Click Administration.  
The Administration Menu appears.
3. In the Administration Menu, click Plumtree Utilities.  
The Plumtree Utilities page appears.
4. Click Edit Default Profiles.  
The Default Profiles page appears. The two profiles listed here are the Default Profile and Guest.
5. Click Guest.
6. To create a gadget, click Add Gadgets to this page.  
The Choose Gadgets page appears. The available gadget folders are listed on this page.
7. To select a gadget in a folder, click the folder.  
The gadgets available in that folder appear.
8. Checkmark the gadget that you want to add to the Guest page. You can select multiple gadgets.
9. Click Add to my Page.  
The gadget is added to the Guest page.
10. Click Finish.  
The Guest page appears.
11. Click Finish Profile Editing.

## Enabling Anonymous Users to View Portal Guest Pages

You can allow anonymous users to access guest pages without logging into the portal. To do this, you must create a policy domain in COREid that uses the COREid None Authentication scheme for the NetPointAnonymous user.

---

**Note:** The COREid None Authentication scheme is a special authentication scheme that COREid provides its customers upon request. The NetPointAnonymous user is a special user that COREid creates for its customers upon request.

---

This allows users to go to the main portal page and view guest pages without being challenged by COREid.

You then lock the NetPointAnonymous account to allow anonymous users to view the guest pages without logging into the Plumtree portal. When a user goes to the main portal page but does not log in, COREid logs in the user as NetPointAnonymous and creates an ObSSOCookie for this anonymous user. The ObSSOCookie is sent to the Plumtree portal but the portal treats the user as a guest because the account is locked. Thus, the user can view guest pages.

### To create a policy domain for guest access

1. Launch the COREid Access Manager and click Create Policy Domain.
2. In the General tab, enter a name and description for the new policy domain.
3. In the Resources tab, select HTTP as the resource type.
4. To use a host identifier, create a host identifier in the Access System Console.

You must add a fully qualified host name as one of the host name variations; for example, plum1.oblix.com.

See the *Oracle COREid Administration Guide Volume 2* for more information.

5. Select the portal URL prefix from the drop-down list or create a new one. For example, /portal.
6. In the Default Rules tab, click Authentication Rule and from the Authentication Scheme drop-down list, select None Authentication.

This allows a user to view guest pages without logging in to Plumtree.

7. In the Policies tab, enter the following information:
  - Name: Enter a name for the policy.
  - Description: Enter a description of the policy.
  - Resource Type: Select HTTP.
  - Resource Operations: Select GET.

- Host ID: Enter the ID of the portal host.
- Resource: Select /portal.
- URL pattern: Enter admin/loginoblix.asp.
- Host identifiers: Enter the host identity.
- Query String: Enter a query string.
- Query String Variable: Enter UserID as the name and 2 as the value.

---

**Note:** The Netscape Web server is case-sensitive. Do not change the case of the query string variable name.

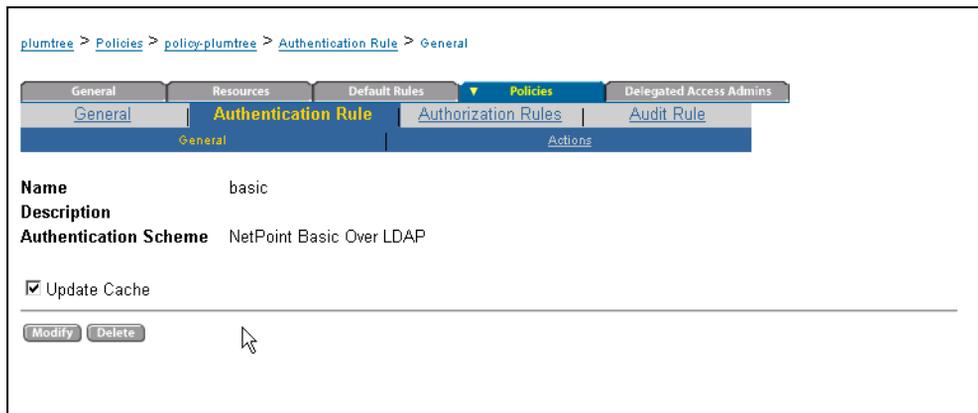
---

8. Click Save to save the policy.
9. In the Policies tab, click the name of your policy.

The policy details page appears.

10. Click Authentication Rule and create an authentication rule.

You can use any authentication scheme such as Basic over LDAP or any custom authentication scheme.



11. In the Authorization Rules tab, select Actions, and add a header variable that you specified in the file oblix.asp.

To facilitate SSO, you must specify this header variable to direct Plumtree to look for this header variable (as described in “Editing Configuration Files to Support SSO” on page 359).

12. In the Return Attribute field, enter the Login ID attribute and click Save.
13. Enable the policy domain
14. On the Plumtree Portal, lockout the user as described “To lock the NetPointAnonymous account” on page 387.

## To lock the NetPointAnonymous account

1. On Plumtree, log in as the administrator to the portal.
2. Click Users and click on the folder that matches the authentication source category you created as your LDAP authentication source.
3. Click the user for the NetPoint None authentication scheme (it will have the prefix of the name you gave to the LDAP authentication source)
4. In the user page, select Lock this User so it cannot be used for login.

The NetPointAnonymous user becomes the equivalent of the Plumtree guest user.

---

**Note:** If you want to log in to Plumtree with the Plumtree database credentials (such as administrator), you must first authenticate yourself on COREid and then click Login as a different User. The Plumtree Login screen appears and you can log in as a different Plumtree user.

To log in to COREid as a different user, first click Logoff to log out of Plumtree and then click Login. The Login box appears, and you can log in as a different user.

---

## Personalizing User Pages

To personalize a user's page, you pass attributes to gadget servers. For example, when a user logs into the portal, a gadget can display a message that greets the user by name. COREid fetches the attributes once, so the gadget itself need not make any additional LDAP calls.

You must configure COREid to pass additional user attributes to the Plumtree portal.

### To personalize user pages

1. Download Plumtree ASP GDK3.5 from the Plumtree Web site (<http://workshop.plumtree.com>) and install it on the machine where you installed the gadgets.
2. Log in to COREid Access Manager and click My Policy Domains.
3. Click the policy that corresponds to the URL prefix that you created; for example, /portal.
4. Select the Default Rules tab, click Authorization Rules and then click the Actions link.
5. Select Add and add another header variable that returns an attribute; for example, a header variable such as FULL\_NAME which returns the attribute cn.

## To pass additional user attributes to the Plumtree portal

1. On the Plumtree Portal server machine, edit the file `userinfo.asp` located in the following directory:

`plumtree_home\4.5WS\PortalPages\PortalPages\sso`

where `plumtree_home` is the location of your Plumtree installation.

2. Log in to the portal as an administrator.
3. Go to Gadgets > General > Hello User.  
The Gadget Editor appears. In the Gadget Editor, you can designate what `UserInfo` settings should be sent to that gadget.
4. Select User Information and check that the required options will be displayed.
5. In the function `PassUserInfo`, add the following code:

```
Dim pUserInfo
Set pUserInfo=ptSession.UserInfo
pUserInfo.FullName = ExtractFromHeaders("headerVarName")
```

where `headerVarName` is the name of the header variable, for example, `FULL_NAME`. This is the variable that the gadget uses to store the attributes that `COREid` passed on to the Plumtree portal.

6. Create a text file named `gadget.asp` and place it in a directory (for example, `Demo`) on a different machine that has a Web server with `WebGate` protection.

Enter the following code into the file:

```
<%
'get the settings object
dim oSettings
set oSettings = Server.CreateObject("GSServices.Settings")
'get the gadget settings
dim dGadgetSettings
set dGadgetSettings = oSettings.GetUserInfoSettings
'if there are no settings, have them set their prefs
if dGadgetSettings.Count = 0 then
%>
Name Not Passed
<%
Response.End
end if
'greet the user by name
%>
<h1>Hello, <%=dGadgetSettings("FullName")%>!</h1>
```

7. Create a virtual directory on your portal server (for example, HelloUserGadget) and map the virtual directory to the directory where you created the file gadget.asp.

This automatically creates a copy of gadget.asp on the virtual directory.

8. Log in to the portal as Administrator and click Administration.

9. Click Gadget Servers and then click Add Gadget Server.

The General Info page appears.

10. Enter the name and description of the Gadget server and click Next.

The Server Settings page appears.

11. In the Base URL field, enter the link to the Web server hosting the gadget and click Next.

A listing of all the gadgets on the server appear in the Gadget Content page.

12. Click Next.

The Security page appears. Accept the defaults and click Finish.

## Embedding COREid Management Functions

You can use gadgets to embed COREid functions in the Plumtree portal. You use portal inserts to create gadgets for any of the user and group management features which are exposed by COREid as portal inserts. For example, you can create gadgets for features such as My Identity, My Groups, User Search, Group Search, Create User, Create Group, and Workflow Ticket Search. For the of list supported portal inserts, please refer to the section on Customizing Portal Inserts in the *Oracle COREid Customization Guide*.

You must synchronize data to update the Plumtree database with any changes that were made to any information that the portal server needs, such as headerVar attributes and login IDs.

You can create several types of gadgets such as iFrames (IE), Pop-up windows (Netscape), or gadgets using gateway.

---

**Note:** The gadget URL must be protected to allow COREid SSO functionality.

---

## Creating iFrame Gadgets

One of the simplest ways to embed COREid functionality in your portal pages is to use COREid Portal Inserts in an iFrame gadget. For example, you can show your identity, your groups profile, or any other Portal Insert feature in an iFrame.

For an example, see the files `gadget.asp` and `adminprefs.asp` located in the directory `IFramework` in the file `SampleGadgets.zip`. You can access `SampleGadgets.zip` at <http://www.oracle.com/support/contact.html>.

---

**Note:** Only COREid portal inserts for the COREid system are supported.

---

### To create an iFrame gadget

1. Move the files `gadget.asp` and `adminprefs.asp` located in the `IFramework` directory to a `IFramework` directory on a webserver-enabled machine.
2. Log on to the Plumtree portal as an administrator.
3. Click Administration > Gadgets > Add Remote Gadget. The General Info page appears.
4. Enter a name and description for the new remote gadget.  
From the drop-down list, choose the gadget server where you placed the files `gadget.asp` and `adminprefs.asp`.
5. Click Next.  
The HTTP Configuration page appears.
6. Enter the following URL for the gadget:  
*virtual\_dir*/iframework/gadget.asp, where *virtual\_dir* is the name of your virtual directory.
7. In the Gateway URL Prefixes field, enter *virtual\_dir*/iframework/ and click Next.  
The HTTP Gateway Caching page appears.
8. Accept the default entries and click Next.  
The Alternative Browsing Devices page appears.
9. Accept the default entries and click Next.  
The Alignment and Administrative Preferences page appears.
10. Select This Gadget Supports Administrative Preferences option and enter *virtual\_dir*/iframework/adminprefs.asp in the Administrative Preferences URL field.

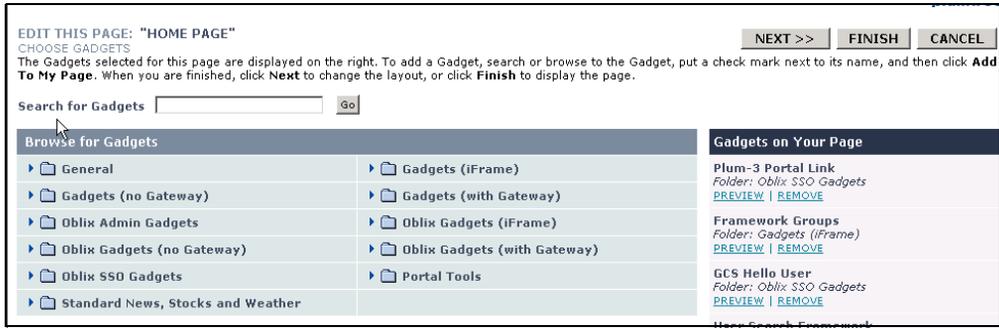
11. Click the link [Click here to edit the administrative preferences](#).  
A pop-up window appears.
12. Enter the portal insert for the COREid identity management functionality of your choice.  
The iFrame gadget will then show the corresponding page in the portal.

---

**Note:** To configure more preferences, edit the adminprefs.asp file.

---

13. To display the gadget in My Pages, navigate to My Pages > Edit this page > Choose gadgets and select an iFrame gadget.



## Pop-Up Window Gadgets

The gadget.asp file and the adminprefs.asp file have the code to detect the browser type and generate either an iFrame or a link to pop-up a window depending on what the browser supports. For a example, see the sample gadget.asp file and adminprefs.asp file located in the iFramework folder in the SampleGadgets.zip file available at <http://www.oracle.com/support/contact.html>.

## Gadgets Used Through a Gateway

If a gadget is used through a gateway, then the portal server is involved as a go-between on all requests between the client and the gadget. This is recommended in cases where you want to access a gadget through the portal, or in cases where the gadget servers are across DMZ.

### To set up a gadget using a gateway

1. Log in as administrator to the portal and click Administration > Gadgets > Add Remote Gadget.  
The General Info page appears.
2. Enter a name and description for the new gadget.
3. From the Gadget Server drop-down list, select the server where the COREid COREid is setup as the gadget server and click Next.

The HTTP Configuration page appears.

4. In the Remote Gadget URL field, enter the URL of the Portal Insert.

For example, to view your profile, the URL would be:

```
host:port /identity/oblix/apps/userservcenter/bin/
userservcenter.cgi?program=view&comp=true
```

where *host:port* is the name of your host server and the port number.

5. In the Gateway URL Prefixes field, enter the URL prefix:  
identity/oblix/apps/userservcenter/bin
6. Accept defaults in all subsequent pages and click Finish.
7. To view the gadget on your page, go to My Pages > edit this page > choose gadgets.

## Importing COREid Dynamic Groups

Typically, COREid is used to determine membership in dynamic groups. The Plumtree database only supports static groups and needs to be able to determine group membership from the LDAP directory without going through COREid. COREid provides the ability to expand dynamic group membership so that information is available to the Plumtree database in the directory.

To update the Plumtree database with dynamic group data from COREid, you create static versions of dynamic groups. To do this, you first expand a dynamic group in COREid. This creates a static version of the dynamic group. You then import the static group into the portal using the Plumtree synchronization job.

You create syndicated gadgets to expand groups and synchronize them with the Plumtree database. Syndicated gadgets use the ExpandGroup portal insert for this function. See the *Oracle COREid Administration Guide Volume 1* for more information.

---

**Note:** You must expand a group before doing a full sync so that the expanded group data is pulled into the Plumtree database.

---

### To import dynamic groups

1. Log in to the portal as the administrator and click Administration.
2. In the Administration Menu, click Syndicated Gadgets > Add Syndicated Gadget.  
The General Info page appears.
3. Enter a name and description for the new gadget.  
Do not select a gadget server.
4. Click Next.  
The HTTP Configuration page appears.
5. In the Remote Gadget URL field, enter the complete URL of the portal insert to expand the group.  
For example, the following code is a portal insert to expand one particular group:

http://<host>/identity/oblix/apps/groupservcenter/bin/  
groupservcenter.cgi?program=expandGroup&groupsToExpand=cn%3  
DBasic%20group1k1%2C%20ou%3DGroups%2C%20ou%3DDealer1k1%2C%2  
0ou%3DLatin%20America%2C%20ou%3DFord%2C%20o%3DCompany%2C%3  
DUS&comp=true

Similarly you can have a portal insert to expand all dynamic groups.

6. In the Basic Authentication User Name field, enter the login ID of the user who has the credentials to expand the group(s).
7. In the Basic Authentication Password field, enter the password of the user.
8. In the Gadget Timeout field, enter the timeout period.

Make sure you give enough timeout period for this gadget, depending on other factors such as the number of groups and users or the number of dynamic groups you have.

**Syndicated Gadget Editor**  
**Syndicated Gadget**  
**HTTP CONFIGURATION**  
Show the portal where to find this Gadget.

Remote Gadget URL: HTTP://SELENIUM.OBLIX.COM:660/

Basic Authentication User Name:

Basic Authentication Password:

Gadget Timeout:

UNC Path to XSL File:

[Customize local gateway directory.](#)

**Gateway URL Prefixes**

[Add a new Gateway URL Prefix](#)

[Click here for help on the HTTP Configuration page!](#)

<< **BACK**    **NEXT** >>    **FINISH**    **CANCEL**

9. Click Next until the Schedule page appears.
10. Select Add this syndicated job to existing job and click the link Add this syndicated job to another job.

A pop-up window appears.

11. Select the original full database synchronization job that you had created and click Finish.

This creates a job that is scheduled to run every time the full sync is run. So you can run one job and have a full LDAP-to-Plumtree database synchronization as well as expand the dynamic groups.

---

**Note:** If you want a specific user's (other than admin) credentials to be used when calling the Expand Group feature, ensure that the user has the rights within COREid to expand the group.

---

### To sync jobs in the right order

1. Log in to the Plumtree portal as the administrator and click Administration > Jobs.

The Job folder appears.

2. Open the folder and click the synchronization job that you created.

The General Info page appears.

3. Click Next.

The Operations page appears. The full sync job and the dynamic group expansion job are listed.

4. Use the green up and down arrows located next to each job to order the group expansion job first and the full synchronization job next.

Job Editor

OblixLDAPAuthnSource Job created on...

OPERATIONS

List the operations that are executed whenever this Job runs.

Actions: Add an Operation

Operations:		
Syndicated Gadget	X	↑
OblixLDAPAuthnSource	X	↓

Set a timeout period for this Job.

Ignore errors and run all Job operations.

[Click here for help on the Operations page!](#)

<< BACK    NEXT >>    FINISH    CANCEL

5. Click Finish.

Whenever this job runs, it will first expand the group in COREid and then do a full synchronization. The Plumtree database is updated with the group membership of the dynamic group.

## Password Management

The password policies set on COREid are always enforced because COREid manages passwords and the COREid Access System evaluates the policies when a user logs in to the portal.

Users who are not logged in to the Plumtree portal must have access to the Lost Password feature. This allows users who have lost their passwords to set a new one.

- To create a Lost Password box, you must use a portal insert to create a gadget that invokes the COREid Lost Password Management feature. Create the gadget as described in “To create an iFrame gadget” on page 390.
- To configure the Lost Password Management feature, you can use the `gadget.asp` and `adminprefs.asp` sample files located in the folder `SampleGadgets.zip > LostPassword` (see “Using Gadgets to Create Integration Features” on page 378).

Users must also be allowed to change their passwords and edit other information. To configure the Change Password feature, you can use the sample files `gadget.asp` and `adminprefs.asp` located in the folder `SampleGadgets.zip > ChangePassword` (see “Using Gadgets to Create Integration Features” on page 378).

See the *Oracle COREid Administration Guide Volume 1* for more information on managing passwords.

---

**Note:** It is recommended that you backup configuration files before you modify them.

---

## Self-Registration

You need to enable, manage, and configure self registration, as follows:

- To enable self-registration, the appropriate self-registration workflow must be created in the COREid User Manager. This workflow is used in the self-registration portal insert.
- To manage self-registration, you must use portal inserts to create a gadget that invokes the COREid Self-Registration feature. The Plumtree database must then be synchronized to be updated with the newly created user. Create the gadget as described in “To create an iFrame gadget” on page 390.
- To configure the Self-Registration feature, you can use the `gadget.asp` and `adminprefs.asp` files located in the folder `SampleGadgets.zip > CreateAccount` (see “Using Gadgets to Create Integration Features” on page 378).

See the *Oracle COREid Administration Guide Volume 1* for more information on enabling self-registration.

## Using the Knowledge Directory

The PlumTree 5.0.4 Knowledge Directory is a portal area that users browse to discover document records containing links to documents that have been uploaded by users or crawlers. Users can add COREid links and COREid protected resources links to the knowledge directory. Whenever a user tries to access links in the knowledge directory, the ObSSOCookie for that user, if it exists, is passed to the resource for user authentication, thus facilitating SSO.

The knowledge directory information is organized into subfolders in a manner similar to file storage volumes and shares. The default portal installation includes a Knowledge Directory root folder with one sub-folder named Unclassified Documents. Before you create additional subfolders, you must define your Knowledge Directory taxonomy, as described in the Deployment Guide for the Plumtree Enterprise Web.

## Setting Knowledge Directory Preferences

You specify how the Knowledge Directory displays documents and folders, including whether to generate the display of contents from a Search Server search or a database query, by setting Knowledge Directory preferences.

### **To set Knowledge Directory preferences:**

1. Click Administration.
2. In the Select Utility drop-down list, click Knowledge Directory Preferences.
3. Specify preferences according to the instructions provided through online help.
4. Click Finish.

## Creating Folders

To create a Knowledge Directory folder, complete the following procedure.

### **To create a Knowledge Directory folder:**

1. Navigate to Directory > Edit Directory.
2. Navigate to the folder into which you want to place a new subfolder.
3. Launch the Folder Editor.
4. Specify a name and description, then click OK.
5. Select the Edit Details icon, then complete the settings according to the instructions supplied in the online help.

## Uploading Documents

To upload documents to the Knowledge Directory folder, complete the following procedure.

### **To upload a document:**

1. Browse to the folder where you want to upload the document.
2. From the Submit a Document drop-down list, choose Simple Submit or choose a data source.
3. Complete the submission forms as described in the online help.

Use filters to control what content goes into which folder. A filter sets conditions to sort documents into associated folders in the Knowledge Directory. Please see PlumTree documentation for more details.

# 7 Integrating Siebel 7 with NetPoint SSO

This chapter describes the integration of NetPoint with the Siebel 7 e-business platform. Siebel 7 is a Web-based suite that combines customer relationship management, partner relationship management and employee relationship management applications.

This chapter covers the following topics:

- “About the Integration with Siebel 7” on page 399
- “Integration Architecture” on page 401
- “Supported Version and Platforms” on page 403
- “Preparing Your Environment” on page 404
- “Setting Up NetPoint SSO for Siebel Application Server” on page 404

## About the Integration with Siebel 7

Integrating NetPoint with Siebel 7 provides a secure Internet infrastructure on which to run identity management for all customer applications and processes. NetPoint integrates identity and access management across Siebel 7, enterprise resources, and other domains deployed on eBusiness networks. NetPoint provides the foundation for managing the identities of customers, partners, and employees across Internet applications. These user identities are combined with security policies for protected Web interaction.

Integrating NetPoint with Siebel 7 provides the following NetPoint features to Siebel 7 implementations:

- **Authentication, Authorization, and Auditing** services for Siebel 7 applications.
- **NetPoint Single Sign-On (SSO)** for Siebel 7 applications and other NetPoint-protected resources. NetPoint provides SSO within a single domain or across multiple domains.
- **NetPoint Authentication Schemes**, listed below, provide single sign-on for Siebel 7 applications.

- **Basic**—Users must enter a username and password in a pop-up window supplied by the Web server. This method can be redirected to SSL.’
- **Form**—This method is similar to the basic challenge method, but users enter information in the custom HTML form. You can choose the information users must provide in the form that you create.
- **X509 Certificates**—X.509 digital certificates over SSL. A user’s browser must supply a certificate.
- **Integrated Windows Authentication (IWA)**—Users will not notice a difference between a NetPoint authentication and IWA when they log on to the desktop, open an Internet Explorer (IE) browser, request a NetPoint-protected Web resource, and complete single sign-on
- **Microsoft .NET Passport**— .NET Passport is a component of the Microsoft .NET framework. The plug-in is a Web-based authentication service that provides single sign-on for Microsoft-protected Web resources
- **Custom**—Additional forms of authentication can be incorporated through use of NetPoint’s Authentication Plug-in API.
- **Session Timeout**—NetPoint allows you to set the length of time that a user session is valid.
- **Ability to Use the NetPoint COREid System for Identity Management**—The COREid System provides identity management features such as portal inserts, delegated administration, workflows, and self-registration to applications such as Siebel 7. NetPoint provides self-registration for new users and customers, providing flexibility in how much access to provide to people upon self-registration. NetPoint’s identity workflows enable a self-registration request to be routed to appropriate personnel before access is granted.

In addition, NetPoint provides self-service functionality, allowing users to update their own user identity profile.

## Siebel 7 Components

The integration with NetPoint SSO involves the Siebel 7 components described below.

**Siebel Gateway Name Server**—The name server provides persistent backing of Siebel server configuration information, including definitions and assignments of component groups and component operational parameters as well as Siebel server connectivity.

**Siebel Database Server**—The Siebel database server contains the data used by Siebel clients.

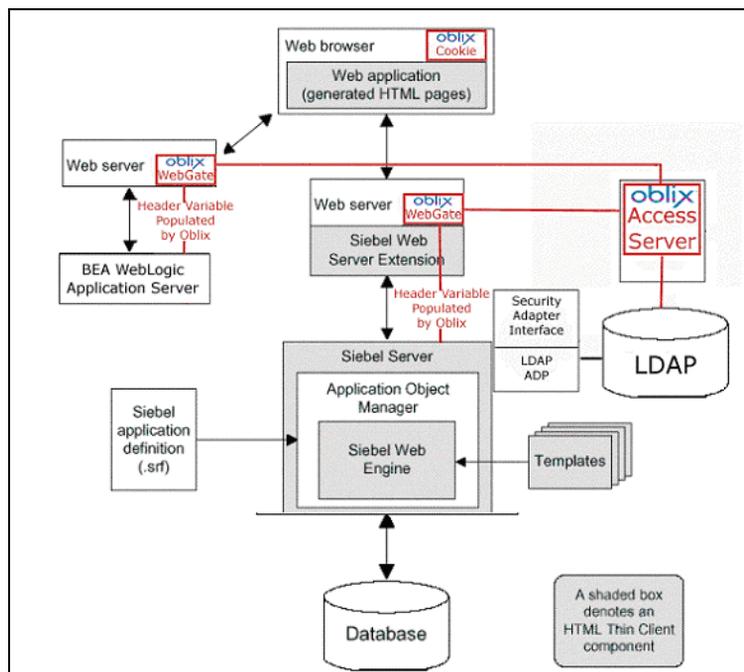
**Siebel Server and Siebel Web Server Extension**—The Siebel Server along with the Siebel Web Server Extensions supports Siebel .COM Web applications and HTML thin clients.

## Integration Architecture

The preferred method of Web single sign-on (SSO) with Siebel 7 is achieved by passing a header variable populated with an attribute value that is stored in the LDAP directory. NetPoint authentication rules permit passing the appropriate HTTP header variable to Siebel 7. The NetPoint WebGate intercepts the user's HTTP request and checks for a session cookie. If the cookie does not exist or it has expired, the user is challenged for credentials. NetPoint verifies the credentials, and if the user is authenticated, the WebGate redirects the user to the requested resource and passes the required header variable to the Siebel application. The Siebel application initiates a session which is kept at the Siebel Web Engine.

Figure 13 illustrates a scenario where the user authenticates to a NetPoint-protected resource and is then granted access to a Siebel 7 application.

**Figure 13** NetPoint Integration with Siebel 7



**Process overview: Authentication with the integration**

1. A user attempts to access content or an application on a server.
2. WebGate intercepts the request.
3. To determine if the resource is protected, WebGate checks the Access Server for a security policy.

The security policy consists of an authentication scheme, authorization rules, and allowed operations based on authentication and authorization success or failure.

4. If the resource is protected, WebGate checks for the user’s session cookie. If a valid session exists, WebGate passes the header variable to the Siebel server. If a valid session does not exist, WebGate prompts the user for credentials.
5. If the credentials are successfully validated, NetPoint executes the actions that are defined in the security policy and sets an HTTP header variable that maps to the Siebel user ID.
6. WebGate redirects the user to the requested Siebel resource.

7. The Siebel application recognizes the NetPoint header variable, authenticates the user, and initiates a session. The header variable is stored in the Siebel Web Engine.

The user can now access any NetPoint protected resource, such as a BEA WebLogic Web application, without being prompted for credentials.

If the user is not authorized, the user is denied access and redirected to another URL determined by the organization's administrator.

## Supported Version and Platforms

The following table illustrates valid NetPoint and Siebel integrations:

	NetPoint 5.2	NetPoint 6.0.1	NetPoint 6.1.1	NetPoint 6.5	NetPoint 7.0
Siebel 7.0.4	.	.	.		
Siebel 7.5.2		.	.	.	.
Siebel 7.5.3		.	.	.	.

NetPoint supports Siebel 7 on the following platforms:

- Windows 2000
- Solaris 2.8

Other supported products for this integration are:

- Microsoft IIS 4.0 and 5.0 for Windows 2000
- Sun (previously iPlanet) Web Server 4.0 and above for Solaris 2.8.
- Microsoft SQL 7.0 or SQL Server 2000
- Oracle 8i or above
- Sun (previously iPlanet) Directory Server 5.1 (for Windows 2000 or Solaris 2.8)

# Preparing Your Environment

Before you can integrate NetPoint with Siebel, you must complete the following steps to prepare your environment.

## **To prepare your environment for integration**

1. Install a supported directory server according to vendor instructions.
2. Install a supported Web server according to vendor instructions.
3. Configure the Web browser to allow cookies according to vendor instructions.
4. Proceed to “Setting Up NetPoint SSO for Siebel Application Server” on page 404.

## Setting Up NetPoint SSO for Siebel Application Server

Setting up NetPoint SSO for Siebel 7 requires the installation and configuration of several Siebel and NetPoint components.

### **Task overview: Setting up NetPoint SSO for Siebel**

1. Install and configure Siebel 7, as described in “Setting Up Siebel 7 for integration with NetPoint” on page 404.
2. Install NetPoint and a WebGate and create NetPoint access control policies to protect Siebel resources, as described in “Setting up NetPoint for Integration with Siebel 7” on page 409.
3. Test the integration, as described in “Testing Integration Between NetPoint and Siebel” on page 410.

## Setting Up Siebel 7 for integration with NetPoint

Setting up Siebel 7 for integration involves the steps below.

### **To setup Siebel 7 for integration with NetPoint**

1. Install the Siebel components below, as described in the Siebel documentation:
  - a) Siebel Gateway Server
  - b) Siebel Server
  - c) Siebel Database Server
  - d) Siebel Web Server Extension

2. Verify that Siebel eBusiness Applications and Web Server Extension are working properly.
3. Ensure that the Siebel client and the Siebel server are able to communicate with each other through TCP/IP, as described in the Siebel documentation.
4. Add at least three users to LDAP:
  - Test
  - The Siebel Anonymous User
  - The Siebel Application User

In addition to your regular users, Siebel uses two additional user accounts from the directory: Anonymous User and Application User. In addition, you will need to create an attribute in regular user accounts for storing the Siebel database user information. See the information on creating users in the directory in the *Security Guide for Siebel eBusiness Applications* for details.

5. Add user records in the Siebel database corresponding to the registered users.
 

You need a record in the Siebel database that corresponds to the test user that you created in the LDAP directory. You also must confirm that the seed data record exists for the Anonymous User for your Siebel customer or partner application. This database record must also match the Anonymous User that you create in the LDAP directory. See the information on adding user records in the Siebel Database in the *Security Guide for Siebel eBusiness Applications* for details.
6. Set the values of the following configuration parameters as shown below.

For more information on Web SSO integration, see Siebel’s *Security Guide for Siebel eBusiness Applications*.

Table 17 describes the parameters to set for the Siebel Name Server configuration file. These parameters are set in the [Object Manager] section.

**Table 17** Name Server Parameters

Parameter and value	Notes
OM - Configuration File = siebel.cfg	This is the name of the configuration file for the application that you are implementing. The OM Configuration File name is application-specific.
OM - Read Configuration From Server Component Parameters = TRUE	
OM - Proxy Employee	Enter PROXYE
OM - Username BC Field	Leave empty.

**Table 17** Name Server Parameters

Parameter and value	Notes
Security Adapter Name = LDAP	The name of the security adapter that you implement, as it appears in the [Security Adapters section in the application configuration file.

Table 18 describes the parameters to set for the eapps.cfg file. This file contains configuration details for the Siebel Web Server Extension component. It is located in the \BIN directory where the Siebel Web Server Extension is installed (C:\sea704\SWEApp). You can add these parameters to either the [Default] section for the Siebel-specific application, for example, [/esales\_enu].

Obliv recommends that these parameters be added to the specific Siebel eBusiness application section.

**Table 18** eapps.cfg Parameters

Parameter and value	Notes
AnonUserName = sadmin	This is the userID of <ul style="list-style-type: none"> <li>• The seed data User record that is provided for the Siebel eBusiness Application that you implement</li> <li>• Or the User record that you create for the anonymous user.</li> </ul> This entry must match the directory entry for the anonymous user.
AnonPassword = sadmin	This is the password that you created in the directory for the Siebel Anonymous User.
SingleSignOn = TRUE	
TrustToken = HELLO	This value can be any string. It is used between Siebel components.
UserSpec = SSO_SIEBEL_USER	The HTTP header variable where the user's identity key is placed for retrieval by the authentication manager.
UserSpecSource = Header	Specifies that the user's identity key is coming from an HTTP header variable.

Table 19 describes the parameters that you specify in the Siebel Application Parameter File (for example, siebel.cfg).

**Table 19** Siebel Application Parameter File

Section where this parameter appears	Parameter and value	Notes
[SWE]	AllowAnonUsers = TRUE	If you do not set this parameter to true, browser looping may occur.
[Security Adapters]	LDAP = LDAP	
[LDAP]	SingleSignOn=TRUE	
[LDAP]	TrustToken = HELLO	Enter the TrustToken that you specified for the same variable in eapps.cfg.
[LDAP]	DllName = sscfldap.dll	
[LDAP]	ServerName = <i>hostname or IP address</i>	Specify the fully qualified host name or the IP address of the LDAP directory server.
[LDAP]	Port = <i>port number</i>	This is the port that the directory listens on.
[LDAP]	BaseDN = " <i>DN</i> "	Specify a DN, for example, "dc=wwm,dc=oblix,dc=com". The Base DN is the top node in the tree where users are stored. Users can be added directly or indirectly below this subdirectory.
[LDAP]	UsernameAttributeType = <i>uid</i>	Enter the attribute in the directory for the Siebel user ID.
[LDAP]	PasswordAttributeType = <i>userPassword</i>	The attribute in the directory where the user's password is stored.
[LDAP]	CredentialsAttributeType = <i>mail</i>	This parameter should be the attribute in the directory that contains the credential.

**Table 19** Siebel Application Parameter File

Section where this parameter appears	Parameter and value	Notes
[LDAP]	Application User = "DN"	Specify the attribute used to store the value for the Siebel database user. Specify the full DN value for the Siebel Application user you created in the directory. Example: "uid=SIEBELAPP, ou=Users, dc=machine, dc=domain, dc=com"
[LDAP]	ApplicationPassword = <i>password</i>	Enter the password that you assigned to the Siebel application user.

### To set the Siebel Name Server Configuration Parameters

1. Log into a Siebel employee application, such as Siebel Call Center, and make one of the following choices from the application-level menu:
  - To set enterprise level parameters, choose View > Site Map > Server Administration > Enterprise Configuration.
  - To set server level parameters, choose View > Site Map > Server Administration > Servers.
  - To set component level parameters, choose View > Site Map > Server Administration > Components.

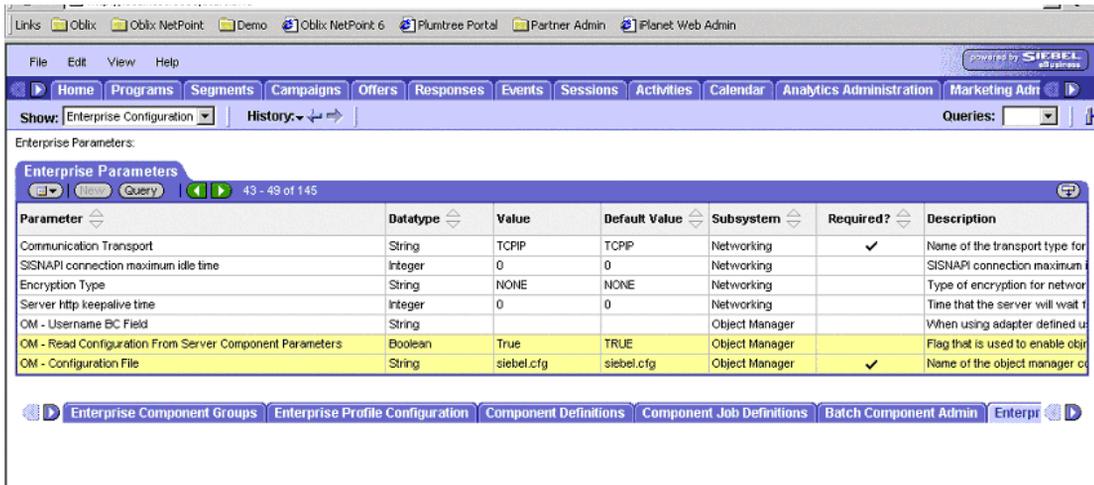
If you are setting parameters at the server or component level:

- To set enterprise-level parameters, click the Enterprise Parameters view tab.
- To set server-level parameters, click the Server Parameters view tab.

- To set component-level parameters, click the Component Parameters view tab.

Because application-level parameters override enterprise level settings, Oblix recommends that you set the Siebel parameters for SSO integration at the application level.

The following screen illustrates setting enterprise-level parameters:



2. Select a parameter record, edit the Current Value field, and then click Save.
3. Restart the Siebel Server to allow the changes to take effect.

## Setting up NetPoint for Integration with Siebel 7

Setting up NetPoint for integration with Siebel 7 involves the steps below.

### To set up NetPoint for the integration

1. Install NetPoint and ensure that you have installed a WebGate on the Web server instance supporting the Siebel Web server extension, as described in *NetPoint Installation Guide*.
2. Synchronize the time on all servers where Siebel and NetPoint components are installed.

Each Siebel application has its own document directory. You can either protect each application individually or protect the higher-level directory under which the applications reside.

3. In NetPoint, create a policy domain to protect Siebel resources on Web servers where Siebel and the NetPoint WebGate are installed, as described in the *NetPoint Administration Guide*.

NetPoint sets header variables that are passed on to the Siebel eBusiness Application to allow access only to specified users.

4. In the Authorization Rule > Actions page of the policy domain protecting the Siebel resource, configure the action to map a NetPoint Header variable uid to the Siebel uid.

---

**Note:** The Header variable set in the NetPoint policy should be equal to the value of the UserSpec parameter in the eapps.cfg file.

---

In the example shown below, the uid is mapped to the SSO\_SIEBEL\_USER HTTP header variable as follows:

**Type**—HeaderVar

**Name**—SSO\_SIEBEL\_USER

**Attribute**—uid

5. In the Authorization Rules > Allow Access page of the policy domain, select the NetPoint/Siebel users to whom you want to grant access to the resources that are protected by the policy domain.

## Testing Integration Between NetPoint and Siebel

After configuring the integration of NetPoint with Siebel, you should test for successful NetPoint authentication and single sign-on with Siebel 7.

The following is a test for single sign-on between a non-Siebel, NetPoint-protected Web page and the NetPoint-protected Siebel Web Server Extension.

### To test NetPoint Single Sign-on

1. Create a demo document directory on an IIS Web server.
2. Add an index.html document to the “demo Web directory on the Web server.
3. Create a NetPoint policy to protect the content of the demo Web directory.  
Require basic LDAP authentication to access the content in this directory.
4. Create a NetPoint policy to protect a Siebel eBusiness application (for example, eMarketing) and require basic LDAP authentication for it.
5. Open a Web browser and enter the URL for the IIS Web server’s main page (<http://hostname>).

The main page is displayed. User authentication should not be required.

6. Access the Siebel eBusiness application URL for the IIS Web server from the same browser used in step 1.

Basic authentication should be required.

7. Access the Siebel eBusiness application URL for the IIS Web server from the same browser used in step 1.  
Access to the Siebel eBusiness application should be allowed. The user should not be challenged for credentials.
8. Close the browser and open a new browser session. Access the Siebel eBusiness application URL for the IIS Web server.  
Basic authentication should be required. After the user enters credentials, the Siebel eBusiness application should be displayed.
9. Access the demo document directory URL for the IIS Web server from the same browser user in step 8.  
Access to the demo document directory should be allowed. The user should not be challenged for credentials.
10. Repeat step 1-step 9 for the Sun ONE Web server.

The following is a test of the NetPoint session timeout.

### **To test NetPoint Session Timeout**

1. Configure the NetPoint session timeout to be five (5) minutes and restart the Web servers.
2. Open a Web browser and the IIS Web server's main page (<http://hostname>).  
The main page is displayed. User authentication should not be required.
3. Access the Siebel eBusiness Application URL for the IIS Web server from the same browser used in step 2.  
Basic authentication should be required. After the user enters credentials, the Siebel eBusiness application should be displayed.
4. Leave the browser window open and idle for more than five minutes.
5. Refresh the browser window using the Refresh button.  
Basic authentication should be required. After the user enters credentials, the Siebel eBusiness Application should be displayed.
6. Repeat step 1-step 4 for the Sun ONE Web server.



# 8 Integrating mySAP Applications with NetPoint SSO

This chapter describes the integration of NetPoint with mySAP.com e-business platform.

This chapter covers the following topics:

- “About Integrating NetPoint with mySAP” on page 413
- “SAP Components” on page 414
- “Integration Architecture” on page 415
- “Preparing to Integrate NetPoint with SAP” on page 417
- “Setting up NetPoint SSO for mySAP” on page 418
- “Integrating the SAP Enterprise Portal” on page 422
- “Configuration Files” on page 426
- “Configuration Examples” on page 426
- “Template Examples” on page 429

## About Integrating NetPoint with mySAP

Integrating NetPoint with mySAP allows the use of NetPoint functionality across all mySAP Web-based applications and other NetPoint-protected enterprise resources and applications.

Integrating NetPoint with mySAP provides the following NetPoint features to mySAP implementations:

- NetPoint single sign-on (SSO) for mySAP applications and other NetPoint-protected resources.
- Authentication, authorization, and auditing services for mySAP applications.
- NetPoint authentication schemes, listed below, to provide single sign-on for mySAP applications:

- Form
- Basic
- Custom
- X509 Certificates
- Integrated Windows Authentication
- Microsoft .Net Passport
- Ability to use the NetPoint COREid System for identity management.

The COREid System provides identity management features such as portal inserts, delegated administration, workflows, and self-registration to applications such as mySAP.

## SAP Components

The integration of NetPoint SSO with mySAP involves the SAP components described in the following sections.

### SAP Internet Transaction Server

SAP Internet Transaction Server (ITS) is a mySAP.com component that provides a Web front-end and allows access to data from the SAP R/3 applications. SAP R/3 provides Enterprise Resource Planning (ERP) functionality for the mySAP.com e-business platform.

SAP ITS consists of two major components: AGate and WGate.

The AGate is responsible for session management including mapping of SAP R/3 screens or function modules to HTML. AGate manages Web sessions including timeout handling and SAP R/3 connection pooling. Based on SAP R/3 information, it generates HTML documents that are forwarded to WGate.

The WGate passes requests to AGate and receives HTML pages back from AGate. The WGate supports various HTTP server interfaces such as Apache, Netscape Server Application Programming Interface (NSAPI), and Internet Server Application Programming Interface (ISAPI).

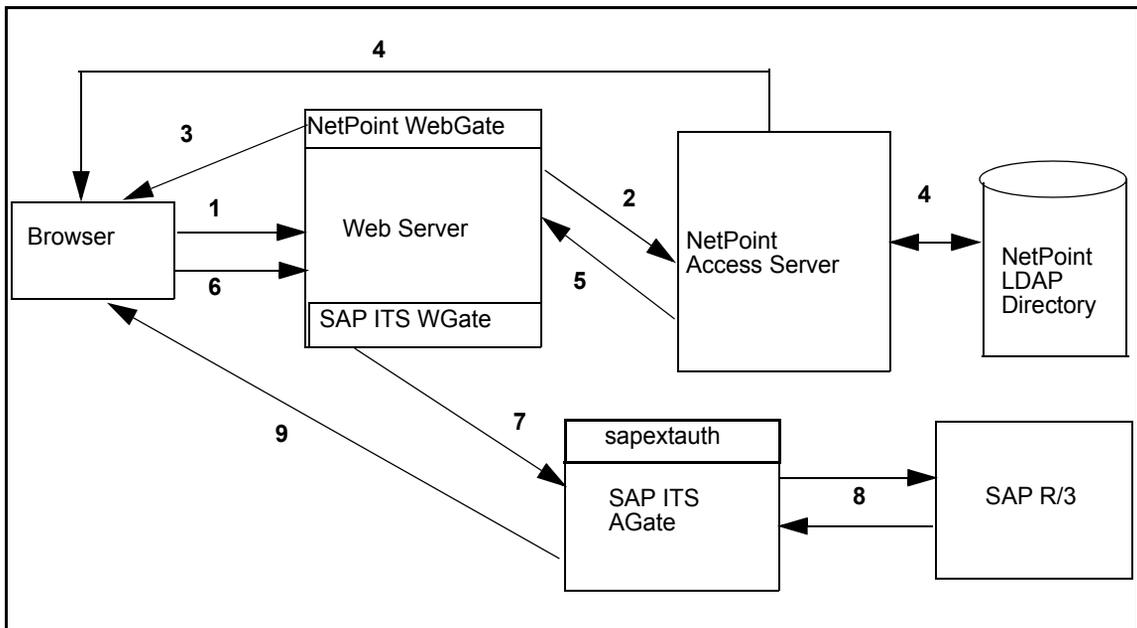
## Pluggable Authentication Service

The Pluggable Authentication Service (PAS) is a part of the Internet Transaction Server that is used for single sign-on between SAP and third-party security providers. PAS allows NetPoint to authenticate users who attempt to access SAP.com resources.

## Integration Architecture

Figure 14 illustrates the integration between NetPoint and SAP ITS. The process overview that follows describes a scenario where the user first authenticates to a NetPoint-protected resource and is then granted access to a mySAP application.

**Figure 14** Integration Between NetPoint and SAP ITS



### Process overview: Integration Between NetPoint and SAP ITS

**1 and 2**—When a user attempts to access content or an application on a company's server, WebGate requests the Access Server for the security policy to determine if the resource is protected.

The security policy defines the required authentication scheme, authorization rules, and allowed operations. Based on the authentication/authorization success or failure, the specified actions are executed.

**3**—Provided that the resource is protected, WebGate prompts the user for credentials required for authentication.

**4**—If the credentials are successfully validated, NetPoint authenticates the user and sets an encrypted SSO Cookie in the user's browser.

**5**—Following authentication, the authentication rules are applied based on the security policy. Subsequent actions are executed.

If the user is authorized, access to the requested content is allowed. If the user is not authorized, the user is denied access and redirected to another URL determined by the organization's administrator.

**6**—The user enters the URL for the NetPoint-specific ITS PAS service.

For integration with mySAP, an ITS-specific HTTP header variable is created and filled with unique NetPoint-SAP R/3 mapped User ID information.

The NetPoint-specific ITS service uses sapextauth module to extract HTTP header variable to identify NetPoint-SAP R/3 mapped User ID.

**8 (optional)**—The SAP Workplace Server maps the external NetPoint UserID to the SAP System ID.

It is recommended that NetPoint extract the correct SAP User ID from the LDAP directory used for the initial authenticated user. In this case no lookup table is required in the SAP system.

**9**—If mapping is successful, the AGate issues the user a SAP Logon Ticket and redirects the user to either Workplace service or any other ITS service. Future ITS URLs will use issued SAP Logon Ticket for passing logon information into the SAP R/3 system.

# Supported Versions and Platforms

NetPoint supports the following versions of SAP Server and the SAP Enterprise Portal:

SAP	SAP Portal
SAP R3 v4.6D SAP ITS v6.10 and v6.20	v5.0 SP2

## Preparing to Integrate NetPoint with SAP

Before you can integrate NetPoint with SAP, you must complete the following tasks.

### To prepare for the integration with SAP

1. Install the following SAP applications:
  - SAP ITS v 6.10 with Patch Level 11, Compilation 4 with Patch Level 340 or later (refer to SAPNet Note: 494984).
  - A SAP ITS component configured to talk to the R3 System through Secure NetWork Communications (SNC). SNC is required to generate SSO2 security tickets.
  - sapntauth library (refer to SAPNet Note: 493107)

Refer to SAP documentation for information on installing SAP applications and components.
2. Install the following NetPoint components:
  - COREid Server
  - WebPass
  - Access Server
  - Access Manager
  - WebGate

See the *NetPoint Installation Guide* for information on installing NetPoint components.
3. For each Web server instance that has ITS installed, install and configure a NetPoint WebGate.
4. Ensure that mySAP and NetPoint components are able to communicate with each other through TCP/IP.

5. Ensure that the servers on which SAP ITS and NetPoint components are installed have a fully qualified domain name.

For example, integrate-1.oblix.net

6. Synchronize the time on all servers where SAP and NetPoint components are installed.
7. Ensure that the users exist in the NetPoint LDAP directory as well as on the SAP R3 system database.

The user ID in NetPoint and SAP must be same or be mapped to each other. Any attribute in a user's profile can be configured as the SAP ID and passed directly to SAP. Alternatively, SAP can be configured to map the SAP ID to any user attribute that it receives from NetPoint.

8. Configure the Web browser to allow cookies.

## Setting up NetPoint SSO for mySAP

Setting up NetPoint SSO for mySAP requires the installation and configuration of several SAP and NetPoint components.

### Task overview: Setting Up NetPoint SSO for mySAP

1. Setup SAP, including the items below, as described in “Setting Up SAP for Integration with NetPoint” on page 419:
  - Install the SAP GUI
  - Install and configure Web server instances for SAP ITS
  - Install SAP ITS
  - Test connections between ITS and SAP R/3
  - Install and configure SAP SNC
  - Configure SAP PAS for NetPoint header variables

---

**Note:** For instructions on installing SAP components, refer to your SAP documentation.

---

2. Setup NetPoint, including the items below, as described in “Setting Up NetPoint for Integration with SAP” on page 420.
  - Install NetPoint WebGate
  - Create NetPoint access control policies to protect SAP resources
3. Complete activities in “Testing Integration Between NetPoint and SAP” on page 421.

# Setting Up SAP for Integration with NetPoint

## To set up SAP for integration with NetPoint

1. Install the SAP Graphic User Interface (GUI) on the client machine.  
This is the Web interface for SAP R/3 applications. It dynamically converts SAP transaction screens to HTML pages.
2. Install and configure two Web server instances; one instance for administrative (ADM) purposes and the second instance as an interface to SAP R/3 applications.

After you have configured the instances, test the connection from ITS to SAP R/3 for both instances.

## To test the ADM instance installation

1. Open a Web browser and enter the URL to log into the ADM instance; for example:

`http://host:port/scripts/wgate/admin/!`

or

`http://host:port/scripts/wgate/adminremote/!`

where *host* is a fully qualified name of the host machine such as *xyz.domain.com*, and *port* is the port number of the host machine.

## To test SAP R/3 instance installation

1. Open a Web browser and enter the following URL to access the GUI of the SAP R/3 instance:

`http://host:port/scripts/wgate/webgui/!`

where *host* is a fully qualified name of the host machine such as *xyz.domain.com* and *port* is the port number of the host machine.

The mySAP.com login screen appears.

2. Install and configure SAP Secure Network Communication (SNC).

SAP SNC provides secure connectivity from the AGate to the SAP R/3 applications. SNC is recommended because NetPoint provides an authenticated user ID to SAP.

If the WGate is installed on a different server than the AGate, it is recommended that you configure SAP SNC between the two servers.

## To set up SAP PAS for integration with NetPoint

1. Configure the SAP PAS system to use Header Variables for SSO:

Configure WGate to pass the NetPoint header variables to AGate. To do this, use the parameter PassHeader located in the wgate.conf file.

For example:

Ex.PassHeader HTTP\_SAPUID

Refer to the SAP documentation for the location of the wgate.conf file.

2. Define the information that PAS requires to use NetPoint as an external authentication provider. To do this, you must configure the PAS Service for NetPoint in the Oblix.srvc file located in

*SAP\_install\_dir*\ITS\2.0\ITSInstanceName\templates

where *SAP\_install\_dir* is the directory where you installed SAP, and *ITSInstanceName* is the name of the ITS instance that you configured.

3. Create and configure PAS templates to handle login, error, and redirect actions that may occur when using NetPoint's authentication service. Save these templates in the *SAP\_install\_dir*\ITS\2.0\ITSInstanceName\templates directory

Create the directory structure and files as follows:

<Name of Service>

<Name of Theme>

login.html

extautherror.html

redirect.html

*Name of Service* is the name of the service file; for example, oblix.srvc.

*Name of Theme* is the name of the theme parameter in the oblix.srvc file.

## Setting Up NetPoint for Integration with SAP

### To set up NetPoint for integration with SAP

1. Install NetPoint WebGate on the Web server instance supporting the ITS connection to the SAP R/3 system.

See the *NetPoint 5.2 Installation Guide* for information on installing a WebGate.

2. In NetPoint, create a policy domain to protect SAP resources under /scripts/wgate.

To do this, create a policy domain that protects the Web servers where SAP ITS and NetPoint WebGate are installed. NetPoint sets header variables that are passed on to the NetPoint-specific ITS service, allowing access only to specified users.

See the *NetPoint 5.2 Administration Guide* for information on creating policy domains.

3. In the Authentication Rule > Actions page of the policy domain, configure the action to set a NetPoint header variable uid to the SAP uid.

The following example maps uid to the SAPUID:

```
HeaderVar HTTP_SAPUIDuid
```

4. In the Authorization Rules > Allow Access page of the policy domain, select the NetPoint/SAP users to whom you want to grant access to the resources that are protected by the policy domain.

The single sign-on configuration is now complete.

## Testing Integration Between NetPoint and SAP

After you have integrated NetPoint with SAP, test for successful NetPoint authentication and single sign-on with mySAP.

The following procedures test the following scenarios:

- A valid login to a SAP R/3 application with a user ID that is authorized both in NetPoint and in SAP.
- A valid login to a SAP R/3 application with a user ID that is authorized in NetPoint but is unauthorized in SAP.
- A valid login to NetPoint COREid System and a SAP R/3 application with a user ID that is authorized in both NetPoint and SAP.
- A valid login to NetPoint COREid System and a SAP R/3 application with a user ID that is authorized in NetPoint but is unauthorized in SAP.

If NetPoint *authentication* has been set up correctly, as an authorized user in both NetPoint and SAP, you will be allowed to access the NetPoint COREid System as well as any SAP R/3 application. If you are an authorized user only in NetPoint, you will be allowed to access only COREid but not a SAP R/3 application.

If NetPoint single sign-on has been set up correctly as an authorized user in both NetPoint and SAP, you will need to authenticate to NetPoint only once. After successful authentication, you will be able to access the NetPoint COREid System and multiple SAP R/3 applications without authenticating again.

### **To test NetPoint authentication**

1. Access any SAP R/3 application.

If integration was successful, NetPoint will challenge you for your credentials.

2. Log in with an authorized NetPoint/SAP user ID.

You will be allowed to access the R/3 application.

3. Attempt to log in to a SAP R/3 application with a user ID that is authorized in NetPoint but is unauthorized in SAP.

Login will fail with message stating that your password is invalid.

---

**Note:** The message is incorrect. The message should state that your user ID is invalid.

---

### **To test NetPoint SSO**

1. Access any SAP R/3 application.

If the integration was successful, NetPoint will challenge you for credentials.

2. Log in with an authorized NetPoint/SAP user ID.

You will be allowed to access the R/3 application.

3. Attempt to log in to NetPoint COREid System.

If single sign-on is successful, you will be able to log in to COREid without being challenged by NetPoint.

4. Attempt to log in to NetPoint COREid System and a SAP R/3 application with a user ID that is authorized in NetPoint but is unauthorized in SAP.

You will be able to log in to COREid but not into SAP. SAP will display a message stating that your password is invalid.

## **Integrating the SAP Enterprise Portal**

A portal provides a single point of access to enterprise data and applications, presenting a unified and personalized view of information to employees, customers, and business partners.

The SAP Enterprise Portal, which runs on top of SAP R/3, provides unified information from enterprise applications, data warehouses, unstructured document collections, and the Internet.

Integrating NetPoint with the SAP Portal provides the following NetPoint functionality:

- Ability to use NetPoint COREid System to manage users and groups.

NetPoint and SAP Enterprise Portal share the same LDAP directory. When a new user or group is created in NetPoint, the SAP user repository is updated with the new data.

---

**Note:** The SAP Portal supports only static groups.

---

- NetPoint SSO for SAP Enterprise Portal and other NetPoint-protected resources.

NetPoint authenticates and authorizes users who attempt to access the SAP Portal. After successful authentication and authorization, users can access any NetPoint protected resource or application without being prompted again for credentials.

## Preparing for SAP Portal Integration

Complete the following tasks before you integrate NetPoint with the SAP Enterprise Portal.

### Task overview: Preparing for SAP Portal Integration

1. Ensure that you have installed SAP and NetPoint versions listed in “Supported Versions and Platforms” on page 417.

Ensure that NetPoint is running.

2. Install the following SAP Enterprise Portal 5.0 SP5 components as described in the SAP documentation:

- SAP J2EE Engine 6.1
- SAP Enterprise Portal 5.0 SP2

## Integrating NetPoint with the SAP Portal

Integrating NetPoint with the SAP Enterprise Portal consists of the following tasks:

1. Configure users in NetPoint.

To do this, you must create a user workflow in NetPoint.

See the *NetPoint 5.2 Administration Guide* for information on user workflows.

2. Configure the SAP Portal.
3. Configure the SAP Portal for external authentication.

This allows you to integrate NetPoint’s authentication, authorization, and single sign-on functionality with the SAP Portal.

4. Configure NetPoint for the SAP Portal.

## **To configure users in NetPoint**

1. Log in to the COREid System as a Master Identity Administrator.
2. Launch the NetPoint User Manager application
3. Create a test user workflow in the NetPoint COREid System.  
A one-step workflow is adequate.
4. Test the new workflow by searching for the test user.
5. Note down the test user's ID and Password.

---

**Note:** If a WebGate is installed on the SAP Portal, you must add the machine's IP and 127.0.0.1 to the IPValidationExceptions parameter list in the webgatestatic.lst file located in *WebGate\_Install\_Dir*\access\oblix\apps\webgate, where *WebGate\_Install\_Dir* is the directory where WebGate is installed.

---

## **To configure the SAP Portal**

1. Log into the Portal as a SAP Portal Administrator
2. Navigate to System Configuration > User Management Configuration and configure the following:
  - Directory server.
  - Authentication server.  
Configure the Authentication server to LDAP and populate the fields accordingly.
  - Portal Roles.
  - Portal Mapping.
3. Restart the Web server and the Application server.

See the SAP documentation for information on configuring the SAP Portal.

## **To configure the SAP Portal for external authentication**

1. Log in to the SAP Portal as a SAP Portal Administrator.
2. Navigate to User Management Configuration and configure the Authentication Server to External and set the User Name Header to UID.
3. Restart the Web server and the Application server.

## **To configure NetPoint for the SAP Portal**

1. Log in to the NetPoint Access Manager as a Master Access Administrator
2. Configure a WebGate.

3. Configure a Host Identifier for the fully qualified machine name; for example, oblixsap.com.
4. Install a WebGate on the SAP Portal machine.
5. Configure a policy domain to protect the following URLs:
  - /SAPPortal
  - /Common Tools
  - /irj
  - /hrnp\$2430001

See the *NetPoint 5.2 Administration Guide* for information on the tasks mentioned above.

6. In the policy domain, specify an authentication scheme of the type Basic Over LDAP; specify the HTTP Header Variable as UID with an LDAP attribute of HTTP\_UID.

This is discussed in “To set up NetPoint for integration with SAP” on page 420.

## Testing Integration Between NetPoint and SAP Portal

After you have integrated NetPoint with the SAP Enterprise Portal, test the integration.

### To test the NetPoint-SAP integration

1. Log in to NetPoint as newly created test user.

If the integration was successful, you will be allowed access to the NetPoint COREid System and the NetPoint Access System.
2. Log into the SAP Portal as the newly created test user.

If the integration was successful, you will be allowed to access to the Portal.
3. In the Portal, navigate to User Mapping and search for other defined users.

# Configuration Files

The oblix.srvc file is located in  
*SAP\_install\_dir*\ITS\2.0\ITSInstanceName\templates directory

where *SAP\_install\_dir* is the directory where SAP is installed and *ITSInstanceName* is the name of the ITS instance. The oblix.srv file contains the PAS service information described in Table 20. PAS requires this information to use NetPoint as an external authentication service.

**Table 20** oblix.srvc Configuration Parameters

Parameter	Description
theme	Specifies the templates to be used when running the Oblix service on SAP.
xgateway	Specifies that an external authentication method is used.
extid_type	Specifies the user type.
remote_user_alias	Specifies the HTTP header variable that will pass the user ID.
redirectHost	Specifies the host in the redirect URL after a successful login.
redirectPath	Specifies the path in the redirect URL after a successful login.
login_to_upcase	Converts the LoginID to uppercase to meet SAP requirements.
client	Name of the client that gets redirected.

## Configuration Examples

This section provides examples of the following mySAP configuration files:

- WGate.config
- Oblix.srvc
- global.srvc

## WGate.config

The WGate.config file is used to pass the NetPoint HTTP header variable to AGate. This is required to enable SSO between SAP ITS and NetPoint.

The following is an example of WGate.config:

---

```
</url>
<url https://_default_:443/scripts/wgate>
    instance          CE6
</url>
<instance ADM>
    <agate>
        Host          ob1x-2ks2
        PortAGate     sapavw00_ADM
        PortMManager  sapavwmm_ADM
        Secure        1
        Type          2
        MultiProcess  no
    </agate>
    TraceFile
        "D:\Program Files\SAP\ITS\2.0\ADMtraces\WGate.trc"
    TraceLevel      1
</instance>
<instance CE6>
    <agate>
        Host          ob1x-2ks2
        PortAGate     sapavw00_CE6
        PortMManager  sapavwmm_CE6
        Secure        1
        Type          2
        MultiProcess  no
    </agate>
    TraceFile
        "D:\Program Files\SAP\ITS\2.0\CE6traces\WGate.trc"
    TraceLevel      1
</instance>
<global>
    PassHeader       HTTP_SAPUID
</global>
```

---

## Oblix.srvc

The `obltx.srvc` file defines the information needed by the PAS to use NetPoint as an external authentication mechanism.

See Table 20, “`obltx.srvc` Configuration Parameters,” on page 426 for a description of its parameters and the location of the file.

The following is an example of the `Obltx.srvc`:

---

<code>~theme</code>	<code>99</code>
<code>~xgateway</code>	<code>sapextauth</code>
<code>~extauthtype</code>	<code>HTTP</code>
<code>~extid_type</code>	<code>UN</code>
<code>~remote_user_alias</code>	<code>~http_http_sapuid</code>
<code>~redirectHost</code>	<code>obltx-2ks2.obltx.com:1080</code>
<code>~redirectPath</code>	<code>/scripts/wgate/webgui/!?!~client=850&amp;~language=EN</code>
<code>~redirectQS</code>	
<code>~redirectHttps</code>	<code>0</code>
<code>~login_template</code>	<code>login</code>
<code>~login_to_upcase</code>	<code>1</code>
<code>~dont_recreate_ticket</code>	<code>1</code>
<code>~client</code>	<code>850</code>
<code>~language</code>	<code>EN</code>
<code>~mysapcomgetsso2cookie</code>	<code>1</code>
<code>~timeout</code>	<code>2</code>

---

## global.srvc

In the Global Service file, confirm the addition of `sapextauth` to the `~xgateways` list. Otherwise, add the `sapextauth` entry to `~xgateways`.

The following is an example of the `global.srvc` file:

---

<code>~routestring</code>	<code>/H/sapgate1.wdf.sap-ag.de/S/3297</code>
<code>~appserver</code>	<code>cpce601</code>
<code>~systemnumber</code>	<code>14</code>
<code>~client</code>	<code>850</code>
<code>~login</code>	
<code>~password</code>	
<code>~language</code>	<code>EN</code>

---

```
~timeout          60
~usertimeout     24
~theme           99
~runtimemode     pm
~cookies         1
~multiinstanceservices 1
~urlarchive      /scripts/sapawl.dll
~urlimage        /sap/its/graphics
~urlmime         /sap/its/mimes
~exiturl         http://oblx-2ks2.oblix.com:1080
~clientcert      1
~hostunsecure    oblX-2ks2.oblix.com
~portunsecure    1080
~hostsecure      oblX-2ks2.oblix.com
~portsecure      443
~xgateways       sapdiag,sapxgwfc,sapxginet,sapxgbc,sapextauth
~xgateway        sapdiag
~sncnamer3       p:CN=CE6, O=SAP-AG, C=DE
~sncqopr3        9
```

---

## Template Examples

This section provides examples of the following mySAP template files:

- Login.html
- extautherror.html
- redirect.html

### Login.html

The following is an example of a login form template.

---

```
`declare fieldEcho, getLanguages in "SAPXJutil.dll";`
<h3>Please log on to the SAP System</h3>
<table>
<tr>
<td>
<form method="post" action="`wgateURL()`">
`fieldEcho()`
<table>
```

```

<tr><td>Service:</td><td>`~Service`</td></tr>
`if (~client=="")`
  <tr><td>Client:</td><td><input name="~client" value="`RSYST-MANDT`"></td></tr>
    <tr><td>
      <input name="~clientinput" type="hidden" value="1">
    </td></tr>
`end`
`if (~language=="")`<tr><td>Language:</td>
  <td>
    <select name="~language">
      `if (getLanguages ("langId", "langDesc") == 0)
        repeat with i from 1 to langId.dim`
          <option value="`langId[i]`">`langDesc[i]`</option>
        `end
      else`
        <option value="en">No allowed languages specified! Using English as
default.</option>
      `end`
    </select>
  </td></tr>
`end`
  <tr><td></td><td>`~MessageLine`</td></tr>
</table>
</td>
</tr>
<tr>
<td>
  <table align=center">
    <tr>
    <td>
      <input type=submit name="~OkCode=/0" value="Logon">
    </td>
    </tr>
  </table>
</td>
</tr>
</form>
</td>
</tr>

```

## extautherror.html

The following is an example of an error message template.

---

```
<H3>Error during authentication process.</H3>
<P>
Either the authentication or the ticket creation failed.
<P>
`if (~messageline != "")`
  The following error occurred: `~messageline`
  <P> The trace files might contain more information about the problem.
`else`
  The error can't be qualified in more detail.
  <p>
  The trace file may contain further information about this error.
`end`
```

---

## redirect.html

The following is a user redirection template, used for redirecting users to a different URL if they fail to authenticate:

---

```
<html>
<head>
<meta http-equiv="refresh" content="0; URL=~ExtAuthRedirectURL`">
</head>
<body>
</body>
</html>
```

---



# 9 Integrating the RSA SecurID Authentication Plug-In

Oblix provides components that interface with RSA Security products to provide native RSA SecurID® authentication for NetPoint-protected resources. This chapter introduces SecurID authentication and the components, requirements, and processes needed to successfully integrate SecurID authentication with NetPoint NetPoint 7.0.

- “About NetPoint and SecurID Authentication” on page 433
- “Supported Versions, Platforms, and Requirements” on page 437
- “SecurID Authentication Scenarios” on page 444
- “Integrating SecurID Authentication” on page 448
- “NetPoint Authentication Plug-In Parameters” on page 469
- “Active Directory Forest Considerations” on page 472
- “Troubleshooting” on page 476

## About NetPoint and SecurID Authentication

NetPoint integrates with RSA components to provide SecurID authentication.

RSA SecurID authentication is based on two factors: *something the user knows* and *something the user has*.

- **Something the User Knows**—This is a secret personal identification number (PIN), similar in concept to a personal bank code PIN. In this case, the PIN may be system generated or personally chosen and registered with the RSA ACE/Server®, which has been renamed to the Authentication Manager.
- **Something the User Has**—This is the current code generated by a hand held device known as a token. Oblix NetPoint supports all RSA SecurID tokens

including RSA SecurID Standard Card, Key Fob, PINPAD Card, and a software-based security token (SoftID) that resides on a user's computer.

The random unpredictable code generated by the token is known as a tokencode. Together, the user's PIN and the SecurID tokencode become the user's Passcode.

The following components are needed for the integration:

- "RSA Components" on page 434
- "Oblix Components" on page 435

See also, "Integration Summary" on page 437.

## RSA Components

During the SecurID authentication process, users must submit their username and passcode using an HTML form. The RSA ACE/Server authenticates the identity of each user through a computer that is registered with the ACE/Server as a client (ACE/Agent). In this case, one NetPoint SecurID Access Server must be registered and set up as an ACE/Agent. See "NetPoint Access Server and ACE/Agent Requirements" on page 440 for more information.

---

**Note:** While the RSA ACE/Server has been renamed to the RSA Authentication Manager, this chapter uses the original naming convention.

---

The RSA ACE/Server compares the tokencode it has generated with the tokencode the user has entered. Tokencodes change at a specified interval, typically 60 seconds. Time synchronization ensures that the tokencode displayed on a user's token is the same code the ACE/Server software has generated for that moment. Authentication is successful when the tokencodes match.

Two-factor authentication provides stronger legal evidence of who performed the task. When properly implemented, the ACE/Server tracks all login requests and operations to reliably identify the user who is responsible for each logged action. See the administration documentation for RSA 5.1 for details about RSA audit trail reports, the automated log database feature, and monitoring activity in real time.

Oblix NetPoint enables integration of SecurID authentication by providing the following:

- The HTML forms required for SecurID authentication operations
- The CGI script required to authenticate users with the RSA ACE/Server
- The SecurID authentication plug-in, `authn_secuid`, required for the NetPoint SecurID authentication scheme

Oblix NetPoint uses and supports the following RSA security features:

- Two-factor SecurID authentication as described above
- RSA BSAFE® SSL-C and BSAFE® Crypto-C
- RC6 encryption for cookies passed between a WebGate and the user's browser
- Optional RSA Keon® Certificate Server and X.509v3 digital certificates (RSA's Keon public key infrastructure (PKI))

---

**Note:** NetPoint supports, but does not require, certificate publishing. SecurID may be used as an authentication scheme for both NetPoint and Keon Web Passport. However, separate authentications must be completed for each product. See the appropriate implementation guide for RSA Keon Certificate Authority for your LDAP directory server.

---

## Oblix Components

The NetPoint COREid System provides the applications you need to manage users, groups, organizations, identity-based workflows, and delegated administration.

- All ACE/Server users must be added to the NetPoint COREid System.
- Delegated Identity Administrators can create a workflow definition to add ACE/Server users to the NetPoint directory. See the *NetPoint 7.0 Administration Guide Volume 1* for more information.

The NetPoint Access System provides policy-based authentication, authorization, auditing, and Web single sign-on. Access System components for SecurID authentication include the Access Manager, Access System Console, Access Server, and WebGate(s) as described below.

- **The Access Manager**—The Access Manager provides the applications for policy management, designation of resources (Web and non-Web) and policy testing. Master Access Administrators define policy domains and Delegated Access Administrators define the resources to be protected by a policy domain.
- **The Access System Console**—NetPoint Administrators and Master Access Administrators use the Access System Console to define authentication schemes, including the SecurID authentication scheme required for this integration, and authorization rules that allow or deny access to resources. The Access system, host identifiers, and master audit settings are also configured in the Access System Console.
- **The Access Server**—The Access Server receives requests from a WebGate and queries authentication, authorization, and auditing rules stored in the

NetPoint directory server. The Access Server returns the authentication scheme, user credentials, and authorization to the requesting WebGate.

The Access Server installation includes the SecurID authentication plug-in, which is a shared library that makes outbound calls to verify the user's authentication credentials against those on the RSA ACE/Server. To accomplish this, one Access Server must be set up as an ACE/Agent. See "NetPoint Access Server and ACE/Agent Requirements" on page 440.

- **WebGate**—WebGate intercepts and forwards HTTP requests for Web resources to the Access Server for authentication and authorization. The WebGate also starts the user's session, creates session cookies, and passes these to the user's browser.

The NetPoint WebGate installation includes the SecurID forms and the CGI script needed to authenticate users with the ACE/Server and to support two special modes of operation. See "Next Tokencode Mode Support" on page 439 and "New PIN Mode Support" on page 439 for more information.

## Integration Summary

NetPoint 7.0 integrates with SecurID ACE Server 5.0 and supports SDI encryption mode.

Table 21 summarizes NetPoint SecurID integration features.

**Table 21** SecurID Integration Summary

Feature	Support for the feature
Authentication method	Native SecurID authentication
New PIN Support	All
Next tokencode support	Yes
Secondary server support	Yes
Location of node secret on Windows client	%windir%\system32
Location of node secret on Unix client	ACE/Agent installation directory
ACE/Agent installation directory	Net OS Agent and Unix Agent
SecurID user specification	Designated users
SecurID protection of administrators	Yes
NetPoint COREid System features and functions	All
NetPoint Access System features and functions	All

## Supported Versions, Platforms, and Requirements

SecurID authentication is supported on the following platforms according to the requirements specified in discussions below.

- “RSA ACE/Server Requirements” on page 438
- “NetPoint Access Server and ACE/Agent Requirements” on page 440
- “NetPoint WebGate Requirements” on page 441
- “Associate Portal Requirements” on page 443

## RSA ACE/Server Requirements

The RSA ACE/Server software provides SecurID identification and authentication of users in the ACE/Server data directory. Data from ACE/Server user records in the directory are validated with the ACE/Server's token records. The ACE/Server's native LDAP support is separate from, yet compatible with, NetPoint.

Netpoint 7.0 has successfully completed all certification criteria and is certified as ACE/Server 5 Ready. The ACE/Server must be installed on one of the platforms in Table 22.

**Table 22** RSA ACE/Server Platform Support

	ACE Server Version	SecurID Web Server Support
Solaris 8	ACE Server 5.1.1	Apache v1.3.27 Sun ONE v6 SP5
Windows 2000 Advanced Server SP4	ACE Server 5.1.1	IIS 5.0
Red Hat Enterprise Linux AS 3.0 <b>Note:</b> RSA does <i>not</i> provide the necessary libraries on Red Hat Linux 3.0 for the SecurID client. As a result, the authn_secuid DSO is not bundled with the package for Linux in COREid 7.0.4 and the COREid 7.0.4 Access Server for Linux does <i>not</i> support SecurID authentication.	ACE Server 5.2	Apache v1.3.27 Sun ONE v6 SP5

The following installation and configuration must be completed before you begin SecurID integration with NetPoint.

### RSA ACE/Server installation and configuration guidelines

- The SecurID ACE/Server software must be installed on a platform noted in Table 22.
- The system time must be correct to prevent the server and client from being out of sync.
- The SecurID tokens or key fobs must be installed, SecurID users must be created on the ACE/Server, and tokens must be assigned.
- Each user name must be mappable through an LDAP filter to a Distinguished Name in the directory.
- A user who authenticates through a RADIUS server must have a profile in the ACE/Server database that provides a list of requirements the user must meet before the ACE/Server challenges the RADIUS user for a passcode.

- An ACE/Server slave and/or replicated ACE/Server can provide failover if the primary ACE/Server is down.

Setting up your RSA ACE/Server, SecurID tokens, and users is outside the scope of this manual. See the installation and administration documentation for RSA ACE/Server 5.1 for details and troubleshooting tips.

As discussed earlier, Oblix supports all RSA SecurID tokens and SecurID authentication. The following modes are also supported.

- “Next Tokencode Mode Support” on page 439
- “New PIN Mode Support” on page 439

## Next Tokencode Mode Support

During authentication, the ACE/Server may direct the user to provide the next tokencode that appears on their SecurID token to prove that they have the assigned token. This operation is known as Next Tokencode mode, which may be triggered by one of the following situations:

- An incorrect Passcode was provided repeatedly during login.  
When a user attempts authentication with incorrect Passcodes four consecutive times, the ACE/Server turns on Next Tokencode mode, as noted in the ACE/Server’s Activity Report. The next time the user successfully authenticates with their correct Passcode, they are challenged for the next tokencode that appears on their SecurID token.
- The ACE/Server requires confirmation of, or synchronization with, the token.  
Even with a correct Passcode, the ACE/Server administrator may set the Next Tokencode mode On to force the user to confirm that they have the SecurID token or to synchronize the token with the ACE/Server.

When Next Tokencode mode is On, the Next Tokencode challenge form is presented to the user immediately following a successful login. See “Next Tokencode Sequence” on page 446 for more information.

## New PIN Mode Support

The token may be in New PIN mode the first time the user logs in or the ACE/Server administrator may enable New PIN mode. New PIN mode requires the user to complete a sequence of forms to define, or have the system generate, a new PIN number. Table 23 provides a description of the New PIN forms and their functions.

**Table 23** Oblix-Provided New PIN Forms and Functions

New PIN Function	Description
------------------	-------------

**Table 23** Oblix-Provided New PIN Forms and Functions

New PIN Query form	Tells the user a new PIN is required. Provides instructions and fields to fill in.
New PIN Confirmation form (system-generated PINs)	Confirms the system-generated PIN for login. Redirects to the resource within 30 seconds
New PIN form (user-generated PINs)	Asks for a username, Passcode, and new PIN.

Each PIN may be:

- Four to eight alphanumeric or numeric characters
- All the same length or of varying lengths
- Defined by the user or by generated the system

See “New PIN Sequence” on page 446 for more information.

## NetPoint Access Server and ACE/Agent Requirements

All SecurID authentication requests must be directed to a single Access Server (also known as the NetPoint SecurID Access Server). The RSA ACE/Agent v5.x, formerly known as the ACE/Client, is an RSA Security program that must be installed on all installed NetPoint Access Servers. However, only one NetPoint Access Server may be registered as an RSA ACE/Agent to perform the authentication dialog with the RSA ACE/Server.

NetPoint Access Server-compatible Web servers installed on the Operating Systems shown in Table 24 will support RSA ACE/Agent software v5.x.

**Table 24** NetPoint Access Server and ACE/Agent v 5.x Platform Support

Solaris 8 Solaris 9	Windows 2000 Advanced Server SP4 Windows Server 2003 Enterprise Edition
------------------------	--

### NetPoint SecurID Access Server guidelines

- All NetPoint Access Servers in the installation must have the RSA ACE/Agent software installed.

---

**Note:** The RSA ACE/Agent software is included with the NetPoint Access Server on Unix systems and must be installed manually on Windows-based Access Servers.

---

- One NetPoint Access Server must be registered as an ACE/Agent Host on the ACE/Server and must have the RSA ACE/Agent software installed to:

- Enable the Access Server to be recognized as an ACE/Server client
- Manage authentication requests from the client to the ACE/Server
- Enforce two-factor authentication and block unauthorized access
- Provide automatic load balancing by detecting replica ACE/Server response times and routing authentication requests accordingly

See “Setting up the Access Server as an ACE/Agent” on page 450 for details.

- The Access Server on Windows systems must have a certificate from the same CA root as the ACE/Server. This is not needed on Unix systems.
- The system time on the client must be correct to prevent the server and client from being out of sync.
- At least one Access Server and one WebGate must be paired for SecurID authentication.

The NetPoint SecurID Access Server may have multiple WebGates that communicate with it; however, all of these WebGates must be configured to communicate with the one NetPoint SecurID Access Server only.

---

**Important:** Failover is not supported for NetPoint SecurID Access Servers. Only one Access Server can complete SecurID authentication.

---

Each Access Server installation includes the SecurID authentication plug-in, `authn_securid`, located in the following directory. For example, on a Windows system:

```
\AccessServer_install_dir\access\oblix\lib\authn_securid
```

This plug-in is required in the SecurID authentication scheme. See “Creating a SecurID Authentication Scheme” on page 458 for details about using the plug-in. See also “NetPoint Authentication Plug-In Parameters” on page 469.

## NetPoint WebGate Requirements

Each WebGate Web server used for SecurID authentication must support and pass header variables to CGI scripts, as shown in Table 25 on page 442 and discussed below.

- Each WebGate that communicates with the NetPoint SecurID Access Server must be configured to communicate with this Access Server only.
- Only NetPoint-provided WebGates are allowed for SecurID authentication. Do not use any other type of AccessGate.
- Lotus Domino Web servers do not pass header variables to CGIs and cannot be used for SecurID authentication.

- Older NetPoint WebGates are compatible and may coexist with NetPoint 7.0 Access Servers. However, encryption schemes differ:
  - Use RC4 as the encryption scheme if you have NetPoint 5.x and NetPoint 7.x WebGates co-existing in the same system.
  - Use RC6 as the encryption scheme if you have NetPoint 6.x and NetPoint 7.x WebGates co-existing in the same system.
  - Use the AES encryption scheme if you have only 7.0 WebGates installed.
- The Perl plug-in or programming language, v5.005\_03, is required on the each NetPoint WebGate host that communicates with the NetPoint SecurID Access Server that validates credentials with the ACE/Server.
- A pointer to the location of Perl is required in the Oblix-provided SecurID CGI script on each WebGate involved in SecurID authentication (SecurID WebGate). See “Setting Up a SecurID WebGate” on page 454 for more information.

**Table 25** NetPoint 7.0 WebGate Web Server Support

Operating Systems	Sun ONE 6.0 SP 5	Sun Java Sys 6.1	Apache 1.3.29	Apache Reverse Proxy Server 1.3.29	Apache Reverse Proxy Server 2.0.48	MS IIS	Domino R 6	IBM HTTP Server 1.3.26	IBM HTTP HTTP/ RP Server 2.0.47	MS ISA 2000 SP1
Solaris 8	X	X	X	X	X		X	X		
Solaris 9	X	X	X	X	X					
Windows 2000 Advanced Server SP4	X	X	X		X	IIS v5.0	X	X		X
Windows Server 2003 Enterprise Edition					X	IIS v6.0				X
Red Hat Linux AS 2.1			X	X	X			X		
AIX 5.1									X	

Included with each NetPoint WebGate installation are the Oblix-provided SecurID authentication forms in the directories below.

*\WebGate\_install\_dir\access\oblix\lang\langTag\securid-forms*  
*\WebGate\_install\_dir\access\oblix\lang\langTag\securid-forms-adforest*

The forms named below are required to validate a user's SecurID credentials and to support New PIN and New Tokencode modes.

- securid-accept-new-pin.html is not used today
- securid-enter-new-pin.html
- securid-new-pin.html
- securid-new-pin-query.html
- securid-next-tokencode.html
- securid-std-login.html

With the exception of a domain name list for the Active Directory Forest that appears on certain forms, the forms in the two directories are the same. See "SecurID Authentication Scenarios" on page 444 to see the forms.

Also included in the WebGate installation is the SecurID CGI script.

## SecurID CGI Script

Each WebGate installation includes a SecurID CGI script in the following directory:

```
\WebGate_install_dir\access\oblix\lang\langTag\securid-cgi
```

During SecurID authentication operations, the WebGate uses the CGI script to present the appropriate SecurID form to the user based on information received from the NetPoint SecurID Access Server and the ACE/Agent that communicates with the ACE/Server.

See "SecurID Authentication Sequence" on page 444 for information on the standard SecurID login form.

See "New PIN Mode Support" on page 439 and "New PIN Sequence" on page 446 for more information about this mode of operation.

See "Next Tokencode Mode Support" on page 439 and "Next Tokencode Sequence" on page 446 for more information about this mode of operation.

## Associate Portal Requirements

An Associate Portal is a WebGate with special capabilities. Associate Portals are supported for SecurID authentication with the following limitations.

- Lotus Domino Web servers do not pass header variables to CGIs and cannot be used for SecurID authentication.
- Lotus Domino Web servers are not supported for Associate Portal Services.

See the *NetPoint 7.0 Installation Guide* for information about installing NetPoint components and services. See the *NetPoint 7.0 Administration Guide* for more information about configuring and using NetPoint components and services.

## SecurID Authentication Scenarios

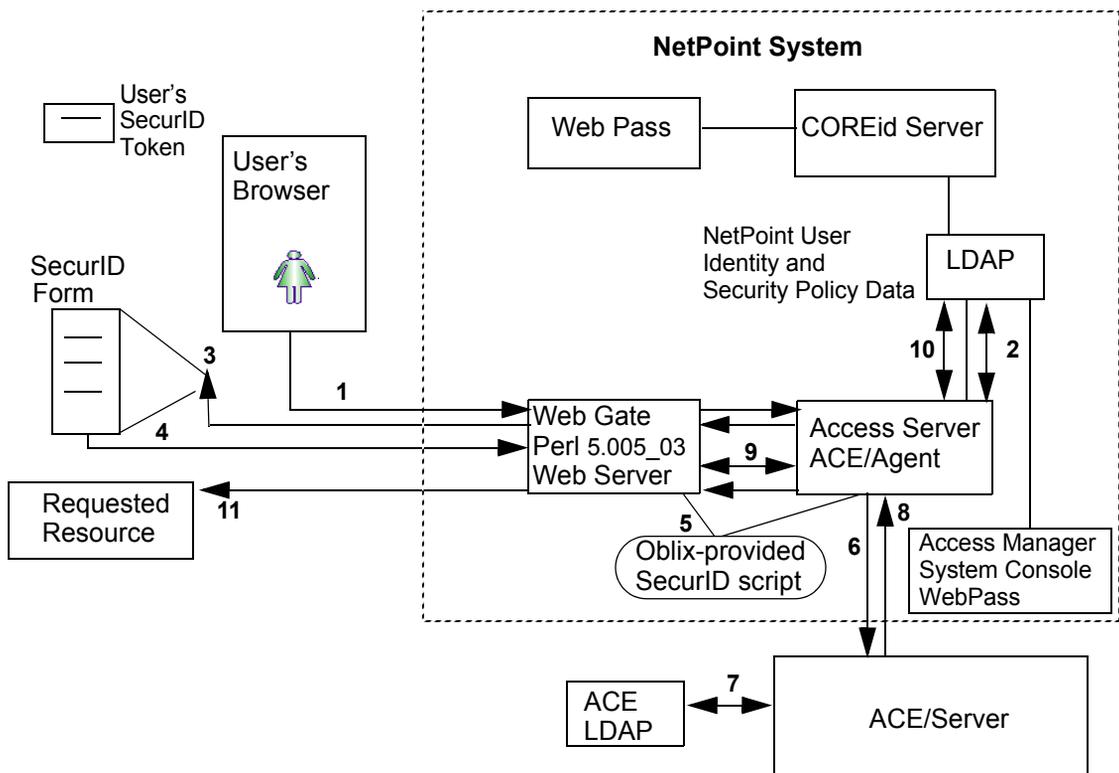
Three scenarios are presented below to cover the three modes of operation:

- “SecurID Authentication Sequence” on page 444
- “Next Tokencode Sequence” on page 446
- “New PIN Sequence” on page 446

### SecurID Authentication Sequence

When a user attempts to access a resource protected by the SecurID authentication scheme, the following process occurs. Figure 15 illustrates the sequence and is followed by a detailed description.

**Figure 15** SecurID Authentication Sequence



## Process overview: When the user requests a resource

1. The WebGate intercepts the resource request and queries the Access Server to determine if and how the resource is protected, and if the user is authenticated.
2. The Access Server queries the directory for the authentication scheme, and receives authentication information from the directory.
3. The Access Server responds to the WebGate, which presents a form challenging the user for a two-part SecurID Passcode, as shown below.

See “Active Directory Forest Considerations” on page 472 to see the forms that include a domain for the Active Directory Forest.



The image shows a screenshot of the RSA SecurID Login Form. At the top, the RSA SecurID logo is displayed. Below the logo, the text 'Login Form' is centered. There are two input fields: 'Username' and 'Passcode'. Below the input fields, there is a blue 'ENTER' button.

4. The user submits credentials to the WebGate.
5. The WebGate presents the credentials to the NetPoint SecurID Access Server.
6. The ACE/Agent on the NetPoint SecurID Access Server performs the authentication dialog and sends an LDAP bind to the ACE/Server.
7. The ACE/Server database matches the SecurID passcode to the user ID and returns a success response to the ACE/Server, which matches the user’s PIN.
8. The ACE/Server returns the response to its Agent, the Access Server. When the user’s credentials are valid, SecurID authentication is successful.
9. The NetPoint SecurID Access Server provides the response to the WebGate. A session is started for the user, so the same authentication method is not required for other Web servers in the domain. The WebGate then queries the Access Server for resource authorization:
  - Under certain conditions a New Tokencode mode may be initiated. See “Next Tokencode Sequence” on page 446.
  - Under certain conditions a New Pin mode may be initiated. See “New PIN Sequence” on page 446.
10. The Access Server queries the NetPoint directory server for authorization information which allows or denies access based upon the authorization rule.

11. When access is granted, the Access Server passes authorization to the WebGate, which presents the resource to the user.

The Master Access Administrator generates a shared secret key to encrypt cookies.

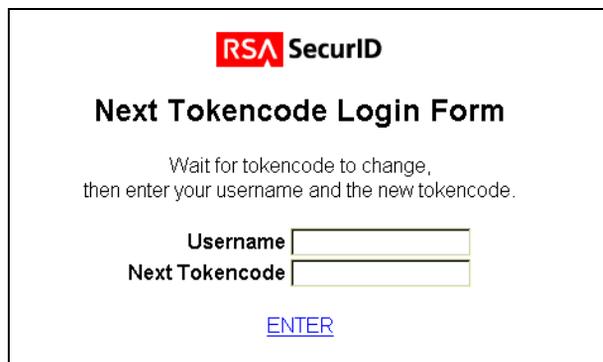
As discussed earlier, two modes may be triggered during the authentication sequence that will alter the user's experience. Standard forms for each mode are shown below. See "Active Directory Forest Considerations" on page 472 for the forms specific to this environment.

## Next Tokencode Sequence

When Next Tokencode mode is On, the user must supply the next tokencode on their SecurID token.

### Process overview: When Next Tokencode is On

1. The WebGate CGI script presents the form below to challenge the user for the next tokencode on the token following a successful login.



**RSA SecurID**

**Next Tokencode Login Form**

Wait for tokencode to change,  
then enter your username and the new tokencode.

Username

Next Tokencode

[ENTER](#)

2. The user enters a username, waits 60 seconds, then enters the next tokencode on the SecurID token.
3. When the tokencode is correct, the Passcode the user originally entered is accepted and the user is authenticated.

See "Next Tokencode Mode Support" on page 439 for additional information.

## New PIN Sequence

When the user is required to have a new PIN, the NetPoint WebGate prompts the user with specific forms.

The sequence is described, and forms are shown, in the following process overviews. See "New PIN Mode Support" on page 439 for more information.

## Process overview: When New PIN mode is On

1. The WebGate SecurID CGI script presents the New PIN Query form to the user, as shown below.



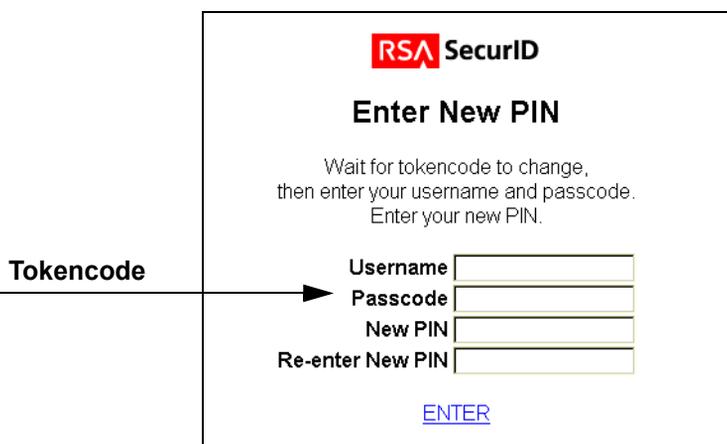
The screenshot shows the RSA SecurID New PIN Query form. At the top is the RSA SecurID logo. Below it is the title "New PIN Query". The instructions read: "You must select a new PIN. Wait for tokencode to change, then enter your username and passcode. Select Yes if you would like the system to generate a PIN for you. If not, select No." There are two input fields: "Username" and "Passcode". Below these is a radio button selection for "System Generated PIN?" with options "Yes" and "No". At the bottom is a blue "ENTER" link.

2. The user waits for the tokencode change, then completes the form and submits it to the WebGate.
3. The WebGate presents this to the Agent on the NetPoint SecurID Access Server for submission to the ACE/Server.

The result is governed by the type of PIN the user requests. In either case, the New PIN process continues.

## Process overview: When the user chooses to define a new PIN

1. WebGate presents the following form so the user can enter the PIN they want.



The screenshot shows the RSA SecurID Enter New PIN form. At the top is the RSA SecurID logo. Below it is the title "Enter New PIN". The instructions read: "Wait for tokencode to change, then enter your username and passcode. Enter your new PIN." There are four input fields: "Username", "Passcode", "New PIN", and "Re-enter New PIN". An arrow labeled "Tokencode" points to the "Passcode" field. At the bottom is a blue "ENTER" link.

2. The user enters a username, then waits 60 seconds and enters the new tokencode and a new PIN to complete the form.

---

**Important:** The user enters the next tokencode, not the passcode.

---

3. The WebGate submits the information to the Agent on the NetPoint SecurID Access Server, which is forwarded to the ACE/Server.
4. The ACE/Server registers the new PIN, which becomes part of the Pincode the user must supply during subsequent logins.
5. The requested resource is provided.

**Process overview: When the user requests a system-generated PIN**

1. The ACE/Server generates a new PIN and the WebGate presents the New PIN confirmation form to the user.



2. The user has 30 seconds to document the new PIN before the confirmation form above is replaced with the form below asking the user to accept the new PIN.
3. The requested resource is provided.

## Integrating SecurID Authentication

Before starting the integration you must complete all prerequisites. After that, you are ready to begin integrating SecurID authentication.

**Task overview: Integrating SecurID authentication**

1. Completing “Preparing Your Environment” on page 449
2. “Setting up the Access Server as an ACE/Agent” on page 450
3. “Setting Up a SecurID WebGate” on page 454
4. “Creating a SecurID Authentication Scheme” on page 458
5. “Protecting SecurID Resources” on page 464
6. “Testing the Policy Domain” on page 468

7. “Adding ACE/Server Users to NetPoint” on page 468

## Preparing Your Environment

Each step below identifies a process that must be completed before you begin integrating SecurID authentication in NetPoint.

### To prepare your environment for SecurID integration

1. Ensure that your RSA ACE/Server installation includes the latest patches and is running properly.

See “RSA ACE/Server Requirements” on page 438 for details. See the installation instructions for RSA ACE/Server 5.1 for details and troubleshooting.

2. Confirm that RSA SecurID user authentication has been properly integrated with your RSA ACE/Server, and add users.

See the installation and configuration instructions for RSA SecurID for details and troubleshooting.

3. Ensure that your NetPoint system is set up and properly running, including the latest patches. Components include:

- COREid Server and WebPass
- Access Manager and Access System Console
- Access Server and WebGate(s)

See the *NetPoint 7.0 Installation* and *NetPoint 7.0 Administration* guides for general information.

4. Install the Perl plug-in or programming language, v5.005\_03, on each WebGate that will communicate with the NetPoint SecurID Access Server.

---

**Note:** The steps below must be completed only when your NetPoint installation includes an Active Directory Forest. The Oblix-provided SecurID forms for the Active Directory Forest include place holders named Domain 1, Domain 2 and Domain 3. These must be changed to valid domain names that accurately reflect your Active Directory Forest installation.

---

### To prepare an Active Directory Forest

1. Set up and ensure that NetPoint works with your Active Directory Forest.

See the *NetPoint 7.0 Installation* for details about installing and deploying NetPoint with Active Directory.

2. Edit the forms to display domain names for your Active Directory Forest. For example:

```
\WebGate_install_dir\access\oblix\lang\langTag\securid-forms-adforest
```

- securid-std-login.html
- securid-nexttokencode.html
- securid-enter-new-pin.html

## Setting up the Access Server as an ACE/Agent

As discussed earlier, this task enables the NetPoint SecurID Access Server to authenticate locally known users with the ACE/Server. The first time a user tries to authenticate through the registered Access Server, a node secret (password between the agent and the ACE/Server) is sent to the Agent in encrypted form and used to encrypt future communications.

---

**Note:** Only one NetPoint Access Server may communicate with the ACE/Server. However, all NetPoint Access Servers require the ACE/Agent software.

---

### **Task overview: Setting up the Access Server as an ACE/Agent**

1. “Registering an ACE/Agent Host” on page 450
2. “Setting up the ACE/Agent Host” on page 451

The information below focuses on successful SecurID integration with NetPoint. Providing complete details about ACE/Agents is outside the scope of this manual. See the administration guide for RSA ACE/Server 5.1 for details about ACE/Agents.

## Registering an ACE/Agent Host

Before you install an ACE/Agent on the NetPoint SecurID Access Server, you must add the server name, IP address, Agent and encryption types to the ACE/Server database. The NetPoint SecurID Access Server must be designated as “open to all locally known users” and Agent Host auto registration should be enabled.

### **To register a NetPoint Access Server as an ACE/Agent Host**

1. Record the name and IP address of the one NetPoint Access Server that will communicate with the ACE/Server for SecurID authentication.
2. Launch the ACE/Server database administration tool on the ACE/Server:
  - **On Unix**—sdadmin
  - **On Windows**—Start > Programs > ACE/Server Database Administration Host

3. Add your Agent Host name, IP address, Agent type, and encryption type.

For example:

**Name**—host\_name

**Network address**—192.168.1.140

**Agent type**—Unix Agent or Net OS Agent

**Encryption Type**—DES

4. Ensure that the Access Server is designated as Open to All Locally Known Users.
5. Ensure that Sent Node Secret is disabled.
6. From the System menu, select Edit System Parameters.
7. Enable the Allow Agent Host Auto-Registration parameter, then confirm changing the system parameters.

---

**Note:** Other parameters do not apply to the NetPoint Access Server Agent.

---

The ACE/Server's `sdconf.rec` file contains settings for all configurable ACE/Server system parameters and Agent Host settings. By default, this file resides in the directory below.

`/ace/data/sdconf.rec`

You will need a copy of this file on the NetPoint SecurID Access Server that communicates with the ACE/Server.

You are ready to set up the NetPoint SecurID Access Server as an ACE/Agent Host.

## Setting up the ACE/Agent Host

The steps in this procedure vary depending upon the platform you are using. See one of the following:

- “On Unix Machines” on page 452
- “On Windows Machines” on page 453

Path names in the examples below may not reflect the actual path names in your environment.

See “NetPoint Access Server and ACE/Agent Requirements” on page 440 for additional information.

## On Unix Machines

On a Unix host, you must copy information from the ACE/Server to the Access Server. Then it is a good idea to verify the ACE/Agent installation on the Access Server, although this verification may be skipped.

### To prepare a Unix-based NetPoint SecurID Access Server

1. Locate and copy appropriate lines from the ACE/Server `%systemroot%/drivers/etc/system` to the Access Server `/etc/system`. For example:

#### ACE/Server `%systemroot%/drivers/etc/system` to Access Server

ACE/Server added to NIS	ACE/Server not added to NIS
securid ...	set shmsys:shminfo_shmmni=100
securidprop ...	set shmsys:shminfo_shmseg=16
adlog ...	set shmsys:shminfo_shmmax=16777216
sdserv ...	set semsys:seminfo_semmni=64
sdadmind ...	set semsys:seminfo_semmnsl=50
sdreport ...	set semsys:seminfo_semmns=100
sdxauthd ...	set semsys:seminfo_semmnu=100
tacacs ...	
radius ...	
radacct ...	

2. Copy or FTP the `sdconf.rec` file from the ACE/Server to the Access Server. For example:

**From ACE/Server**—`/ace/data/sdconf.rec`

**To Access Server**—`/AccessServer_install_dir/access/oblix/config/secrid/sdconf.rec`

### To verify the ACE/Agent installation on the Unix-based host (optional)

1. Locate and start the Agent installation program on the Access Server.

```
./sdsetup -agent [-p path]
```

2. Review configuration information and the Agent Host address field to confirm that you entered the correct hostname in the Access Server `/etc/hosts` file.

The first time the client is used for authentication, the node secret will be copied into a file named `securid` in the `VAR_ACE` directory on the client. Typically this is stored in the default path `/NetPoint/access/oblix/config/secrid`.

3. Ensure the VAR\_ACE environment variable is set properly.

If it is not set properly, the node secret will not be copied the first time the client is used for authentication.

See the Unix installation instructions for the RSA ACE/Agent for additional information and troubleshooting tips.

### **On Windows Machines**

On the Windows-based NetPoint SecurID Access Server registered as an ACE/Agent host, you must create a root certificate to define the encryption protocol between the Agent and the ACE/Server and copy the `sdconf.rec` file to the Access Server before you install the ACE/Agent software on the Access Server. Complete the following procedures:

- Complete the first procedure, below, for the NetPoint SecurID Access Server registered as an ACE/Agent host.
- Complete the second, ACE/Agent installation, procedure on every NetPoint Access Server.

### **To prepare a Windows-based NetPoint SecurID Access Server**

1. Install the ACE/Agent Certificate Agent utility on the Access Server using the ACE/Server CD.
2. Start the ACE/Agent Certificate utility on the Access Server: Start > Programs > ACE Agent > ACE Agent Certificate Utility.
3. Create a root certificate for the machine and make a note of where this is stored.

New\_Root  
Certificate and keys  
**Host name**—host\_name  
**Organization**—Name  
**Country**—US

A certificate file is created for this host and, by default, is stored in the following directory path.

`\Program Files\SDTI\ACE Agent Certificate\host_name.crt.`

## To install the ACE/Agent on each Windows-based Access Server

1. Copy or FTP the sdconf.rec file from the ACE/Server to the Access server.

For example:

**From**—D:\ace\data\sdconf.rec

**To**—C:\%systemroot%\system32\sdconf.rec

You are ready to install the ACE/Agent software on the Access Server. A Windows client requires ACE/Agent .dlls in the \WINNT\system32 directory. The minimum Agent installation is all you need to obtain the appropriate files: aceclnt.dll and sdmsg.dll.

2. Run agent.exe from the ACE/Server CD and select Common shared files and User documentation.
3. Specify the path to the root certificate you created earlier:  
*\Program Files\SDTI\ACE Agent Certificate\host\_name.crt*
4. Give the path to the sdconf.rec file that you copied to this machine:  
*\%systemroot%\system32\sdconf.rec*
5. Repeat step 1 through step 4 on each Access Server in your installation to include the aceclnt.dll and sdmsg.dll required for authn\_secuid plug-in initialization.

When you have the Access Server set up as an ACE/Agent you are ready to set up the SecurID WebGate(s).

## Setting Up a SecurID WebGate

Before you can use SecurID authentication, you must set up the SecurID WebGate Web server to successfully locate and use the Oblix-provided SecurID forms and the CGI script.

### Task overview: Setting up a SecurID WebGate

1. “Relocating Oblix SecurID Directories” on page 455
2. “Setting up the SecurID CGI Script” on page 455
3. “Configuring the CGI Directory” on page 456

---

**Note:** You must repeat the three procedures in this task on each WebGate used for SecurID authentication and ensure that each WebGate communicates with only the one NetPoint SecurID Access Server.

---

## Relocating Oblix SecurID Directories

Successful SecurID authentication requires that the three Oblix-provided securid directories installed with the WebGate are located in a directory that is configured as the Web server's document directory. This can be:

- The primary document root  
or
- A virtual document root

Unless the WebGate was installed under the Web server's document root, you must relocate a copy of the Oblix-provided securid directories.

### To relocate the Oblix-provided SecurID directories

1. Locate the three securid directories on your WebGate host. For example, on a Windows system:

```
\WebGate_install_dir\access\oblix\lang\langTag\securid-cgi  
\WebGate_install_dir\access\oblix\lang\langTag\securid-forms  
\WebGate_install_dir\access\oblix\lang\langTag\securid-forms-adforest
```

---

**Note:** If the three SecurID directories are in the Web server's primary document or a virtual document root directory, skip to "Setting up the SecurID CGI Script" on page 455. Otherwise, complete step 2.

---

2. Copy the three securid subdirectories under a Web server document directory.

For example:

```
\iPlanet\WS6sp4\docs\Netpoint\securid-cgi  
\iPlanet\WS6sp4\docs\Netpoint\securid-forms  
\iPlanet\WS6sp4\docs\Netpoint\securid-forms-adforest
```

Later, when you create a SecurID authentication scheme, you may need to adjust your scheme challenge and authn\_securid plug-in parameters accordingly. See "Creating a SecurID Authentication Scheme" on page 458.

## Setting up the SecurID CGI Script

To operate properly, the Oblix-provided SecurID CGI script must point to the correct location of Perl v5.005\_03 on the WebGate.

As shown in the example below, the three SecurID directories were copied from their original installation directory to the Web server's document root using the previous steps.

## To define the path to Perl

1. Open the Oblix-provided securid.pl script.

For example:

```
\iPlanet\WS6sp4\docs\Netpoint\securid-cgi\securid.pl
```

2. Ensure that the first line points to the correct location of Perl on this WebGate.

For example:

```
#!/usr/bin/perl -w
```

Next, you will set up the CGI directory on the WebGate Web server.

## Configuring the CGI Directory

The Oblix-provided SecurID CGI directory must be configured as a CGI directory on the Web server. This process will vary depending on your Web server platform:

- **On IIS Web servers**—You need only ensure that the Oblix-provided script is an executable.
- **On iPlanet Enterprise Web servers**—You need only configure the CGI directory as a Web server CGI directory.

**On Unix**—Specify a Programs/CGI directory

**On Windows**—Specify a Shell/CGI Directory

- **On Apache Web servers**—You must modify the httpd.conf file to:
  - Add the Oblix-provided CGI script to the AddHandler section, or uncomment it.
  - Add ExecCGI to a <Directory> container that applies to the directory where the Oblix-provided script is located.
  - Add PassEnv lines outside all <Directory> and <VirtualHost> containers.

See the documentation that accompanies your specific Web server for more information.

## To configure the CGI script on IIS Web servers

1. Locate the securid.pl file.
2. Make the securid.pl file an executable.

See the *Microsoft IIS Web Server Administration* documentation for details about converting the script to an executable.

## To configure a CGI directory on the iPlanet Enterprise Server

1. Log in as the Web Server Administrator, then select your server name and click Manage.
2. Select the Virtual Server Class tab, click the Manage button, and then click the Programs tab.
3. Select the CGI directory for your platform. For example:

**Unix**—Programs/CGI Directory

**Windows**—ShellCGI Directory

4. Add the URL prefix and the full path to the CGI directory, as shown in the Windows example below:

**URL prefix**—NetPoint\securid-cgi

**CGI directory**—C:\iPlanet\WS6sp4\docs\NetPoint\securid-cgi

See the documentation for the iPlanet Web Server Administration Server for more information.

## To configure Apache Web servers for the SecurID CGI script

1. Locate the httpd.conf file and add the following information to the AddHandler section, or uncomment the line if it is already there.

```
AddHandler cgi-script .pl
```

---

**Note:** Be sure to include the space in AddHandler cgi-script .pl; otherwise, the Web server won't start.

---

2. Find a <Directory> container that applies to the directory where you have stored the securid.pl file, and add “ExecCGI” to the end of the line that reads “Options Indexes FollowSymLinks MultiViews,” as shown below:

```
Options Indexes FollowSymLinks MultiViews ExecCGI
```

3. Add the following lines outside all <Directory> and <VirtualHost> containers.

```
PassEnv HTTP_COOKIE
PassEnv HTTP_REDIRECTURL
PassEnv HTTP_FULLFORMDIR
PassEnv HTTP_HTTPTYPE
PassEnv HTTP_NEWPINRETURN
```

See the Apache Web Server documentation for additional details.

Next you will create a NetPoint SecurID authentication scheme that uses the Oblix-provided SecurID plug-in.

## Creating a SecurID Authentication Scheme

This section provides the following topics.

- “Background” on page 458
- “Defining an Authentication Scheme for SecurID” on page 461

Even if you are familiar with NetPoint authentication plug-ins, you may want to focus on specific SecurID requirements presented earlier.

Some information offered in the examples below should be replaced with the appropriate information for your environment.

### Background

The NetPoint Access System protects resources according to policy domains. Each policy domain identifies the resources to be protected and must include one and only one authentication rule. That rule, which is considered the default authentication rule, must contain an authentication scheme which specifies the challenge method used to obtain credentials from the user. Each authentication scheme can include one or more plug-ins to perform additional processing. For Smart Card authentication, you must use the Client Certificate authentication scheme.

A policy domain can include policies to protect resources within the domain in a different or more specific way. Each of these policies can have its own authentication rule, but one is not required. If an authentication rule is not configured for a policy, the default authentication rule for the policy domain applies.

Until the Master Access Administrator delegates administration rights to a policy domain, he or she is the only person who can access that policy domain.

The form below is used to define the authentication scheme. This is available in the Access System Console, Access System Configuration Authentication Management function.

An authentication scheme name is required. The description is optional. The security level definition is the same for all authentication schemes. For more information about authentication schemes, see the *NetPoint 7.0 Administration Guide Volume 2*.

Discussions below provide additional details for SecurID:

- “SecurID Challenge Method” on page 459
- “SecurID Challenge Parameters” on page 460
- “SecurID Authentication Scheme Plug-Ins” on page 461

## SecurID Challenge Method

Each authentication scheme requires a challenge method to obtain user credentials for authentication. Only one challenge method is allowed per authentication scheme.

SecurID requires the form-based challenge method, which means that the user must complete an HTML form during the authentication process. Form-based authentication schemes can pass authorization actions in header variables but cannot pass authentication actions in header variables.

The Basic challenge method does not support SecurID Pincodes, Next Tokencode Mode, nor New PIN Mode.

See the *NetPoint 7.0 Administration Guide Volume 2* for more information about authentication scheme challenge methods.

---

**Note:** Do *not* protect a challenge form or any of its components, such as .gifs and links.

---

## SecurID Challenge Parameters

SecurID requires four challenge parameters to identify what will occur when a user logs in. The four challenge parameters are action, passthrough, creds, and form.

- **action**—You can use the action parameter to present a form to authenticate the user after receiving an initial request for a resource. For SecurID authentication, the form action is initiated by the Oblix-provided CGI script.

Default location of the SecurID CGI script:

```
\WebGate_install_dir\access\oblix\lang\langTag\securid-cgi\securid.pl
```

Relocated to: *\iPlanet\WS6sp4\docs\NetPoint\securid-cgi\securid.pl*

- When the script is installed on a single Web server instance, the relative path is sufficient.
- When the script is on a different Web server instance, the full URL path is required.
- **passthrough**—Passthrough is set to No by default. For SecurID authentication, set passthrough to Yes to pass the login credentials to a post-processing program.
- **creds**—The creds parameter identifies all fields used for login in the HTML forms, in a space-separated list. The parameters needed for SecurID authentication must correspond to fields in the SecurID authentication HTML forms and SecurID plug-in parameters. For example:

```
login username password passcode choice  
choice_label_such_as_system-generated newpin  
PIN_entered_by_the_user newpin2 PIN_re-entered_by_the_user
```

The creds challenge parameter for the user ID should match the userid specified in the credential\_mapping plug-in that is required with SecurID authentication. For example:

**Challenge Parameter**—creds:login

**credential\_mapping plug-in parameter**—  
obMappingFilter="(&(...userid=%login%))"

- **form**—The form parameter indicates where the standard login HTML form is located relative to the Web server's document directory.

For example, when the full path is:

```
\iPlanet\WS6sp4\docs\NetPoint\securid\securid-forms\securid-std-login.html
```

The form parameter is: `\securid-forms\securid-std-login.html`

## SecurID Authentication Scheme Plug-Ins

Two plug-ins are required in the SecurID authentication scheme, `authn_securid` and `credential_mapping`. Each plug-in defines how information will be looked up in the directory server. Again, examples below illustrate concepts and may not portray your environment.

The `authn_securid` plug-in authenticates the user's SecurID credentials against their credentials on the ACE/Server. The two mandatory parameters include `fullformdir` and `machine`.

- **fullformdir**—This parameter identifies the full and complete path from the Web server root to the authentication form directory. For example:  

```
fullformdir=C:\iPlanet\WS6sp4\docs\NetPoint\securid-forms  
fullformdir=C:\Webserver_root\NetPoint\securid-forms-adforest
```
- **machine**—This parameter identifies the fully qualified machine name, including the domain name and port of the WebGate Web server instance that will communicate with the NetPoint SecurID Access Server.

`machine=host_name.domain.com:port`

The `credential_mapping` plug-in maps the user-provided information to a valid Distinguished Name (DN) in the NetPoint directory. Ensure that the `userid` you specify matches the `creds` parameter you specified in the challenge parameter.

```
obMappingBase="o=company,c=us"  
obMappingFilter="(&(objectclass=...orgperson)(...userid=%login%)"
```

A number of parameters are available with each plug-in. See “NetPoint Authentication Plug-In Parameters” on page 469 for more information. With these considerations in mind, you are ready to define the SecurID authentication scheme.

## Defining an Authentication Scheme for SecurID

Only a Master Access Administrator may create authentication schemes. The steps below walk you through the process you must complete to define a SecurID authentication scheme. Differences for an Active Directory Forest are noted where appropriate. See also “Active Directory Forest Considerations” on page 472.

In the example below, the action URL points to the new location of the `securid-cgi` directory as discussed in “Relocating Oblix SecurID Directories” on page 455. Path names may differ in your environment.

### To define the SecurID authentication scheme

1. From the Access System Console, click Access System Configuration > Authentication Management
2. Click the Add button at the bottom of the panel.

3. Enter a name, optional description, and security level for your SecurID authentication scheme.

For example:

**Name**—SecurID Authentication

**Description**—This scheme requires a user to enter a SecurID username (login) and passcode. This scheme also handles Next Tokencode mode and New PIN mode.

**Level**—An integer between 1 and 5 defines the security level.

4. Select Form as the challenge method and enter challenge parameters for this authentication scheme.

For example:

**Challenge Method**—Form

**Challenge Parameters**—

- action:\NetPoint\securid-cgi\securid.pl
- passthrough:yes
- creds:login password choice newpin newpin2
- **form excluding Active Directory Forests**—  
form:\iPlanet\WS6sp4\docs\NetPoint\securid-forms\securid-std-login.html
- **form for Active Directory Forests only**—  
form:\Webserver\_root\NetPoint\securid-forms-adforest\securid-std-login.html

5. Select No beside SSL Required.

If you have more than one WebGate/Access Server pair, redirect to a WebGate that communicates with the dedicated NetPoint SecurID Access Server. Use the fully qualified machine name. Syntax:

`http://host_name.domain.com:port/`

---

**Note:** When you have only one WebGate/Access Server pair, leave the Challenge Redirect field blank and skip to step 8.

---

6. Enter a challenge redirect, as needed for your environment.
7. Save.

System Configuration System Management Access System Configuration

General Plugins Steps Authentication Flow

### Details for Authentication Scheme

<b>Name</b>	SecurID Authenticaion
<b>Description</b>	This scheme requires a user to enter a SecurID username (login) and passcode. This scheme also handles Next Tokencode mode and New PIN mode
<b>Level</b>	1
<b>Challenge Method</b>	Form
<b>Challenge Parameter</b>	action\NetPoint\securid-cgi\securid.pl passthrough:yes creds:login password choice newpin newpin2 form\iPlanet\WS6sp4\docs\NetPoint\securid-forms\securid-std-login.html
<b>SSL Required</b>	No
<b>Challenge Redirect</b>	
<b>Enabled</b>	No

Modify Back

Next, you will create a customized challenge scheme using the `authn_securid` and `credential_mapping` plug-ins.

The `authn_securid` plug-in should be the first. Be sure that the machine parameter includes the fully qualified domain name and port or the host identifier or one of its aliases as specified in the Access System Console. See “SecurID Plug-In Parameters” on page 469 and “Credential Mapping Plug-In Parameters” on page 471 for more information.

- Click the Plug-Ins tab, the Modify button, then select Custom Plugins from the drop down list and specify parameters.

For example:

- Plug-in Name**—`authn_securid`

**Plug-in Name Parameters (excluding Active Directory)**—  
`fullformdir="c:\iPlanet\WS6sp4\docs\NetPoint\securid-forms",`  
`machine="host_name.domain.com:port"`

**Plug-in Name Parameters for Active Directory**—  
`fullformdir="c:\Webserver_root\NetPoint\securid-forms-adforest",`  
`machine="host_name.domain.com:port"`

9. Select the credential\_mapping plug-in from the drop down list and specify parameters.

For example:

- **Plug-in Name**—credential\_mapping

**Plug-in Name Parameters (excluding Active Directory)**—

```
obMappingBase="o=company,c=us",  
obMappingFilter="(&(objectclass=...orgperson)(...userid=%login%))"
```

**Plug-in Name Parameters for Active Directory**—

```
obMappingBase="%domain%", obMappingFilter="(?(objectclass=user)  
(samaccountname=%login%))"
```

10. Check Update Cache to have this take effect immediately, then click Save.

11. Restart the Access Server to load the plug-ins.

You have finished creating a SecurID authentication scheme that will appear in the Authentication Scheme list when you assign rules to a policy domain. See the *NetPoint 7.0 Administration Guide Volume 2*.

---

**Important:** Before you use this scheme, the form's action URL must be protected by a NetPoint policy domain and the action challenge parameter of the form scheme must match the form action URL.

---

## Protecting SecurID Resources

Before you can use the SecurID authentication scheme in a policy domain, you must protect the Oblix-provided SecurID CGI script specified in the action URL of your SecurID authentication scheme. Once protected, you may use the scheme in new policy domains to protect other resources with SecurID authentication.

---

**Important:** As shown below, when protecting the SecurID CGI script you may use any authentication scheme **except** the SecurID authentication scheme. Also, do not protect the forms or any elements in the forms (.gifs, for example).

---

### Task overview: Protecting Securid Resources

1. "Creating a Policy Domain" on page 465
2. "Adding a Resource to Your Policy Domain" on page 466
3. "Defining Rules for this Domain" on page 466

## Creating a Policy Domain

The key to creating an effective policy domain is to group the content that you want to manage in the same way. Each policy domain is defined using the Access Manager and each policy domain includes a definition of:

- Resources to protect
- Schemes, rules, and optional policies (exceptions) for protection
- Administrative rights, optional

For example, you need one policy domain to protect the SecurID authentication script and action URL for the Scheme. This cannot be protected by the SecurID authentication scheme. You will want another policy domain to protect resources using the SecurID authentication scheme.

The following procedure shows how to create a policy domain to protect the SecurID CGI script. Following each sequence is a brief example of a second policy domain that will use the SecurID authentication scheme to protect other resources. The information provided is a sample to illustrate concepts.

The following procedure requires specific policy domain management rights. See the *NetPoint 7.0 Administration Guide Volume 2* for details.

### To create a policy domain to protect the SecurID script

1. Launch the NetPoint Access System Console:

`http://host_name.domain.com:port/access`

2. Select the Access Manager.

The My Policy Domains page appears with functions on the left and current policy domains on the right.

3. Click Create Policy Domain on the left.

The General tab is highlighted and the Name field is active.

4. Enter a name and an optional description for the new policy, then save it.

For example:

**Name**—SecurID-script

**Description**—NetPoint-provided

---

**Note:** The policy domain cannot be enabled until you add a resource type.

---

## Adding a Resource to Your Policy Domain

Only a Master Access or NetPoint Administrator may add resources to a policy domain. When NetPoint is initially installed, no resources are defined. Your environment may include resources.

A resource may be either static or dynamic content.

- Static content includes HTML pages, .gifs, .pdfs
- Dynamic content includes scripts, applications, EJBs

For this integration, you will add the securid-cgi as a resource for the policy domain to protect the Oblix-provided SecurID script.

The administrator who created the first policy domain set the existing root URL used as a base for the policy domain. You may append a different region to the same prefix to define a new URL prefix that is available to other policy domains.

Again, sample specifications to protect a resource with the SecurID authentication scheme are included with the steps below. However, you *cannot* use the SecurID authentication scheme until you protect the SecurID script.

### To add a resource to your policy domain

1. From the Access Manager policy domain page, click the Resources tab then click the Add button and add a resource.

For example:

**Resource Type**—http

**URL Prefix (and region)**—/ NetPoint/securid-cgi

**Description**—Optional

2. Ensure that Update Cache is enabled, then save.
3. Click the Save button and review the information you supplied.

### Defining Rules for this Domain

All Administrators may create an authentication expression for a policy domain or policy. An existing authentication scheme must be specified as the building block.

As discussed earlier, you must protect the SecurID authentication script specified in the action URL of the authentication scheme before you can use the scheme.

When completing the steps below, skip to step 2 if the Default Rules tab is available.

## To define who has access

1. Click Access Manager > My Policy Domains > Securid-script.
2. Click the Default Rules tab, then click the Add button.
3. Enter the details and confirm that you are using the NetPoint Basic Over LDAP authentication scheme.

For example:

**Name**—Securid Authentication

**Description**—Optional

**Authentication Scheme**—NetPoint Basic Over LDAP

4. Ensure that Update Cache is enabled, click the Save button, then review the information summary.

When protecting the SecurID CGI script, you do not want to deny access to anyone. When protecting the SecurID CGI script, allowing everyone enables the Access Server to check each user's credentials with the ACE/Server. By default, nobody is authorized. Allowing access should take precedence.

5. Click the Authorization Rules tab, choose Oblix Authorization Scheme from the drop down list, then save.

For example:

**Authorization Scheme**—Oblix Authorization Scheme

6. Enter a name, an optional description, enable this scheme, ensure that Allow takes precedence, then save.

For example:

**Name**—SecurID allow\_all

**Description**—Optional

**Enabled**—Yes

**Allow takes Precedence**—Yes

In every case, including SecurID CGI script protection, you must specify the Role of those being granted access to this policy domain. For your environment, you may want to grant access only to specific users or groups.

7. Click the Allow Access tab, click the Add button, fill in the form, and save.

For example, to grant access to anyone from the root directory down:

**Role**—Anyone

**Rule**—ldap:///

**Update Cache**

Though not required to protect the SecurID CGI script, you may apply one or more policies to fine-tune access control for the protected resources and apply auditing rules to record access requests and resource use. For these activities, you must have authorization granted by an Access Administrator.

8. Click the General tab and enable the policy domain.

The policy domain is active and the resource is protected. In this case, the SecurID CGI script is protected by the NetPoint Basic Over LDAP authentication scheme and ready to use in policy domains that protect your resources.

9. Repeat “Protecting SecurID Resources” on page 464 to protect your own resources using the SecurID authentication scheme.

See the *NetPoint 7.0 Administration Guide Volume 2* for details about policy domains.

## Testing the Policy Domain

The best way to help you identify and resolve any problems that might arise is to test the policy domain and check various log files to ensure that everything is working properly. See the appropriate manuals for your systems for details.

### To enable logging and testing

1. Enable logging on your Web servers to help track any anomalies during operation.
2. Enable logging your NetPoint SecurID Access Server.
3. Enable logging on your NetPoint SecurID WebGate or WebGates.
4. Enable logging on your RSA ACE/Server to report activity in real time or to create an activity log, as usual.
5. Test your policy domains and Web single sign-on, as usual, to ensure that all are working as expected.

## Adding ACE/Server Users to NetPoint

You must add ACE/Server users to the NetPoint directory to enable access to the protected resource after the user is authenticated.

One way to do this is to use the NetPoint COREid System User Manager application to create a workflow definition. Creating a workflow for this purpose is no different than creating a NetPoint workflow to add other users, as long as you include the following attributes as a minimum.

- Name
- Last Name

- Login
- Password

See the *NetPoint 7.0 Administration Guide Volume 1* for details about creating a workflow definition.

## NetPoint Authentication Plug-In Parameters

SecurID authentication requires the SecurID plug-in and the Credential Mapping plug-in. Each plug-in provides a number of parameters to direct how information is looked up in the directory server.

### SecurID Plug-In Parameters

The parameters below apply to the `authn_secuid` plug-in when defining the SecurID authentication scheme in NetPoint. You may customize the parameter name according to the rules specified in the comments in Table 26. These parameters are case sensitive.

**Table 26** `authn_secuid` Plug-In Parameters

Parameter Name	Default Value	Status	Comments
<code>httpType</code>	<code>http://</code>	Optional	If the webgate doing the SecurID authentication is in SSL, the value should be changed to <code>https://</code> by passing additional parameter <code>httpType="https://"</code>
<code>fullformdir</code>	<none>	Mandatory	<p>This is the full path to the SecurID forms used for authentication, from the Web root to the directory that contains the forms. The value requires a trailing slash. For example:</p> <pre>fullformdir="C:/iPlanet/WS6sp4/docs/NetPoint/secuid-forms/"</pre> <p>or, for Active Directory</p> <pre>fullformdir="C:\Websserver_root/NetPoint/secuid-forms-adforest/"</pre> <p>By default, the forms directories are installed as shown below and should be moved to the Web server's document directory:</p> <pre>\NetPoint\webcomponent\access\oblix lang\langTag\secuidxxx</pre>

**Table 26** authn\_securig Plug-In Parameters

Parameter Name	Default Value	Status	Comments
machine	<none>	Mandatory	<p>This is the fully qualified domain name and port number of the Web server instance that will communicate with the NetPoint SecurID Access Server. For example:</p> <p>machine="machine.domain.com:port"</p> <ul style="list-style-type: none"> <li>• This name must match the host identifier specified in the Access System Console or one of its aliases.</li> <li>• If you are redirecting all SecurID authentications, this should be the Web server name that you are redirecting to.</li> </ul>
formdir	access/oblix/securig-forms	Optional	<p>This is the relative path to the SecurID forms and requires a trailing slash.</p> <p><b>Note:</b> If you customize this value, you must also change it in the Challenge Parameter, form, and SecurID plug-in parameter, fullformdir. In other words, if you place the SecurID forms anywhere other than:</p> <p><i>webserver_docroot/access/oblix/securig-forms</i></p> <p>you <i>must</i> pass the formdir parameter to the authn_securig plug-in with the value appropriately changed. The new value should be the location of the forms relative to the <i>doc_root</i>. Also, the value should <i>not</i> include a trailing slash.</p> <p><b>Also,</b> if the securig cgi script is not accessible at <i>webserver_docroot/access/oblix/securig-cgi/securig.pl</i>, you <i>must</i> edit the various SecurID html forms to point to the correct location. You need to change the following text in each form to point to the correct location of the script.</p> <p>action="/access/oblix/securig-cgi/securig.pl"</p>
username	login	Optional	<p>If you are using the sample forms, set this value to: login. If you customize this value, you must also change it in the Challenge Parameter, creds and also in the credential_mapping obMappingFilter.</p>
passcode	password	Optional	<p>If you are using the sample forms, set this value to: password. If you customize this value, you must also change it in the Challenge Parameter, creds.</p>
choiceLabel	choice	Optional	<p>This is the name of the field in the HTML form corresponding to the user's choice of how a PIN is generated. If you customize this value, you must also change it in the Challenge Parameter, creds.</p>

**Table 26** authn\_securid Plug-In Parameters

Parameter Name	Default Value	Status	Comments
newpinLabel	newpin	Optional	This is the name of the field in the HTML form corresponding to the new PIN entered by the user. If you customize this value, you must also change it in the Challenge Parameter, creds.
newpinLabel2	newpin2	Optional	This is the name of the field in the HTML form corresponding to the new PIN that is re-entered by the user. If you customize this value, you must also change it in the Challenge Parameter, creds.

## Credential Mapping Plug-In Parameters

This plug-in maps the userID to a valid distinguished name (DN) in the NetPoint directory. You can configure the attribute to which the userID is mapped to find the DN. The most common attribute that is mapped to is uid. However, it is possible to map the userID to a profile attribute other than uid by changing the obMappingFilter parameter. See Table 27 below. These parameters are case-sensitive

**Table 27** Credential\_mapping Plug-In Parameters

Parameter	Usage Rule	Description
obMappingBase		Defines the Base DN in the LDAP search.  If omitted or empty, the directory base is used.
obMappingFilter	Mandatory	Defines the Filter in the LDAP search.  This parameter allows use of the obMappingFilter term to filter for categories of end users.
obdomain	Needed with Active Directory Forests	Authenticates a user against an Active Directory Forest when the challenge method is Basic.
EnableCredentialCache		Turns off the credential mapping cache in the credential_mapping plug-in.  You may want to turn off the cache if the same userid may be mapped to different DNs. See the <i>NetPoint 7.0 Developer Guide</i> for more information.

Two subordinate parameters may be used with the obMappingFilter. These parameters can only be used with the obMappingFilter parameter. See Table 28.

**Table 28** obMappingFilter Subordinate Parameters

Parameter	Description
obuseraccountcontrol	When this parameter is activated, or if there is no value, two categories of end users are filtered out: <ul style="list-style-type: none"><li>• Those who have been added but not yet activated in the directory.</li><li>• Those who have been deactivated but remain in the directory.</li></ul>
obEnableCredentialCache	Turns off the credential mapping cache in the plug-in to deactivate the user the next time they authenticate.

## Active Directory Forest Considerations

The following information can be found throughout discussions in this chapter. They are repeated here for quick reference.

### Prerequisites

Before integrating SecurID authentication with an Active Directory Forest, you must complete the steps below.

#### Task overview: To prepare your environment

1. Complete all tasks in “Preparing Your Environment” on page 449 to set up the RSA ACE/Server, SecurID tokens and users, and NetPoint Identity and Access systems.
2. Set up and ensure that NetPoint works with an Active Directory Forest.  
See the *NetPoint 7.0 Installation Guide* and *NetPoint 7.0 Administration Guide Volume 1* for details about installing and deploying NetPoint with an Active Directory Forest.
3. Edit the following forms to replace the place holder domain names with actual domain names for your Active Directory Forest installation. For example, `\WebGate_install_dir\access\oblix\lang\langTag\securid-forms-adforest`.
  - securid-std-login.html
  - securid-nexttokencode.html
  - securid-enter-new-pin.html

## Integrating SecurID with an Active Directory Forest

Any differences for an Active Directory Forest are included below for quick reference and are embedded in discussions elsewhere in this chapter. This section assumes you have completed the tasks in the following sections:

- “Setting up the Access Server as an ACE/Agent” on page 450.
- “Setting Up a SecurID WebGate” on page 454.

---

**Note:** In the example below, the action URL points to the securid-cgi directory as shown in “Relocating Oblix SecurID Directories” on page 455. Path names may differ in your environment. Bold indicates the items that are different for an Active Directory Forest.

---

### To integrate SecurID authentication

1. Follow the instructions in “Creating a SecurID Authentication Scheme” on page 458 with the following changes for the Active Directory Forest.

Changes for Active Directory Forest Challenge parameters:

- **form**—  
`\Webserver_root\NetPoint\securid-forms-adforest\securid-std-login.html`

#### Changes for Active Directory Forest plug-ins and parameters:

- **Plug-in**—`authn_securid`  
**Parameters**—  
`fullformdir="c:\Webserver_root\NetPoint\securid-forms-adforest",machine="host_name.domain.com:port"`
- **Plug-in**—`credential_mapping`  
**Parameters**—  
`obMappingBase="%domain%", obMappingFilter="(?(objectclass=user)(samaccountname=%login%))"`

2. Select Update Cache, then Save to complete your SecurID authentication scheme.
3. Restart the Access Server to load the plug-in.
4. Complete the tasks described in the following sections:
  - “Protecting SecurID Resources” on page 464.
  - “Testing the Policy Domain” on page 468.
  - “Adding ACE/Server Users to NetPoint” on page 468.

## SecurID Forms for an Active Directory Forest

As you can see on the following Oblix-provided forms, the only difference between standard forms for SecurID authentication and the forms provided for SecurID authentication with an Active Directory Forest is the inclusion of the Domain name on the Login form, Next Tokencode form, and New PIN Entry form.

The form in Figure 16 includes the Domain list. See “SecurID Authentication Scenarios” on page 444, for the standard forms.

**Figure 16** Standard SecurID Login Form for Active Directory Forest



**RSA SecurID**

### Login Form

Username

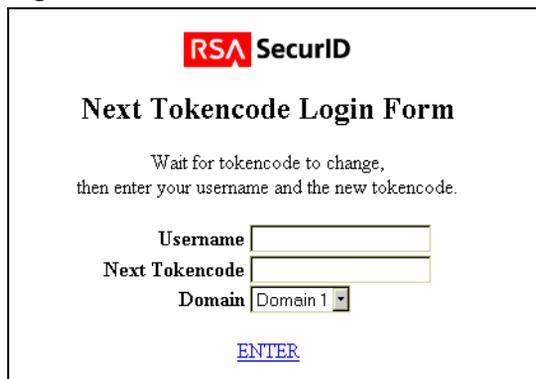
Passcode

Domain

[ENTER](#)

The form in Figure 17 includes the Domain list. See “Next Tokencode Sequence” on page 446 for the standard form and a description of the sequence of events that occurs in this mode.

**Figure 17** Next Tokencode Form for Active Directory Forest



**RSA SecurID**

### Next Tokencode Login Form

Wait for tokencode to change,  
then enter your username and the new tokencode.

Username

Next Tokencode

Domain

[ENTER](#)

See “New PIN Sequence” on page 446 for the standard forms for the New PIN sequences and a description of the events that occur with this sequence. Figure 18, the New PIN Query form, below is slightly different.

**Figure 18** New PIN Query



The screenshot shows a web form titled "New PIN Query" from RSA SecurID. The form contains the following elements:

- RSA SecurID** logo at the top.
- New PIN Query** title.
- Text: "Selection of a new PIN is required." followed by "Wait for tokencode to change, then enter your username and passcode. Select yes if you would like the system to generate a PIN for you, otherwise select no."
- Input fields for **Username** and **Passcode**.
- Radio buttons for **System Generated PIN?** with options  Yes and  No.
- A blue [ENTER](#) button at the bottom.

### **User-Defined PIN Form**

If the PIN was defined by the user, they are challenged by the form and asked to enter their:

- Username
- Tokencode (shown in the form as Passcode)
- New PIN
- Active Directory Domain

The form in Figure 19 includes a Domain list.

**Figure 19** New User-Specified Pin Validation Form for Active Directory

Forest



### Enter New PIN

Wait for tokencode to change,  
then enter your username and passcode.  
Enter your new PIN.

Username

Passcode

Domain

New PIN

Re-enter New PIN

[ENTER](#)

Tokencode →

## Troubleshooting

Following is a brief list of some of the things to check if SecurID authentication is not working as expected.

- “ACE/Agent Issues” on page 476
- “ACE/Server Configuration File” on page 477
- “CGI Directory on SecurID WebGates” on page 477
- “Environment Variable on Unix Systems” on page 477
- “Form-Based Authentication” on page 477
- “Access Server Log” on page 478
- “Permissions” on page 479
- “SecurID Plug-In Parameters with Modified HTML Fields” on page 479

### ACE/Agent Issues

If an “unable to resolve” or “Error:gethostbyname failed” message appear in the CLIENT ADDRESS field when configuration data is displayed, you probably entered the client hostname incorrectly in the /etc/hosts file. In this case, you must edit the host entry to solve the problem.

If the authn\_securid plug-in fails to initialize in an environment with multiple Windows-based Access Servers, verify the status of each Access Server as discussed in the following procedure.

## **To verify the status of each Windows-based Access Server**

1. Confirm that each Windows-based Access Server in your environment has the RSA ACE/Agent software installed.

See “Setting up the Access Server as an ACE/Agent” on page 450.

2. Confirm that only one Access Server is registered with the RSA ACE/Server as the NetPoint SecurID Access Server.

See “Registering an ACE/Agent Host” on page 450 and “Setting up the ACE/Agent Host” on page 451.

## **ACE/Server Configuration File**

The RSA ACE/Server `sdconf.rec` file is required on the NetPoint SecurID Access Server before you can install the RSA ACE/Agent on the Access Server.

This RSA ACE/Server file contains all configurable items for the RSA ACE/Server, including Agent Host specifications.

You must copy this file to the Access Server that will validate user credentials with the ACE/Server before you add the ACE/Agent to the Access server. See “Setting up the Access Server as an ACE/Agent” on page 450 for details.

## **CGI Directory on SecurID WebGates**

Ensure that the `securid-cgi` directory is set up properly on the WebGate.

### **Task overview: Testing the securid-cgi directory**

1. Unprotect a different CGI in the same directory.
2. Access the unprotected CGI to be sure you set up the CGI directory properly.
3. See “Setting Up a SecurID WebGate” on page 454 for information.

## **Environment Variable on Unix Systems**

When setting up your NetPoint SecurID Access Server as an ACE/Agent host, ensure the `VAR_ACE` environment variable is properly set on your Unix system. See “Setting up the Access Server as an ACE/Agent” on page 450 for more information.

## **Form-Based Authentication**

Ensure that form-based authentication is set up properly.

If this the first time the `authn_securid` plug-in has been configured, you must restart the Access Server to load the plug-in.

## Access Server Log

If an authentication plug-in returns an error, it is logged in the Access Server log. You configure this in the Access System Console.

### To set up the Access Server log

1. From the Access System Console, select Access System Configuration > Access Server Configuration
2. Select the server name from the List of All Access Servers, then click Modify.
3. Set Debug On and enter a file name.
4. Restart the Access Server.

## Web Server Logs

These can provide many clues as to what is going wrong. Be sure the enable logging on your Web server.

See the documentation for your Web server for details.

## RSA ACE/Server Logs

If communication has been established between the Access Server and ACE/Server, the sdadmin tool provides access to logs under the Report menu. Both Activity and Exception reports may give you helpful information.

### To verify the ACE/Server log configuration

1. Confirm that you have added the user and assigned a token using the ACE/Server Administrator tool, sdadmin.
2. Verify that you have copied the sdconf.rec file to the Access Server before installing the ACE/Agent.

See “Setting up the Access Server as an ACE/Agent” on page 450 for more information.

3. Locate the file below in the Web server’s document root directory to ensure that the shared secret was downloaded on the first connection between the NetPoint SecurID Access Server and the ACE/Server. For example:

```
\\Planet\WS6sp4\doc\NetPoint\securid
```

## Permissions

Permissions can sometimes cause problems.

---

**Note:** Do not protect the securid.pl script on the WebGate(s). Do not protect the SecurID forms or their directories.

---

### **Confirm that the following permissions are appropriate**

- On the SecurID CGI script, securid.pl
- On the SecurID HTML forms
- On all files
- On the page you are trying to reach

## **SecurID Plug-In Parameters with Modified HTML Fields**

If you have modified the HTML field names in the HTML forms, make sure you have modified the SecurID plug-in parameters to match.



# 10 Integrating NetPoint with Smart Card Authentication

NetPoint 7.0 supports Smart Card authentication with Active Directory and IIS Web servers in homogeneous Windows® environments.

The following discussions explain how to implement Smart Card authentication:

- “About Smart Card Authentication” on page 481
- “About NetPoint Components” on page 482
- “Integration Architecture” on page 483
- “Supported Versions and Platforms” on page 485
- “Setting Up Smart Card Authentication” on page 485
- “About Policy Domains for Smart Card Authentication” on page 490
- “Client Certificate Authentication Schemes” on page 491
- “Troubleshooting” on page 494

## About Smart Card Authentication

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**—This is the user’s secret personal identification number (PIN), similar in concept to a personal bank code PIN.
- **Something the User Has**—This 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.

Smart Card authentication can be used with NetPoint to protect resources. After setting up your environment, Smart Card authentication is triggered when you:

- Insert your ActivCard containing a public key certificate previously issued by the Enterprise Certification Authority (CA) into the reader attached to your computer.

- Request access to a resource protected by the NetPoint Client\_Certificate authentication scheme before inserting your ActivCard into the reader.

The first method displays a window prompting you for your PIN, rather than requesting a username, password, and domain. The second method displays a window prompting you to insert the ActivCard and provide your PIN.

---

**Note:** When you initialize a Smart Card, you are asked to supply a PIN. If the PIN is incorrectly entered three times, the card locks. To restore a locked certificate, either use the unlock code provided during Smart Card initialization or re-initialize the card.

---

## About NetPoint Components

The NetPoint COREid System provides the applications you need to manage users, groups, organizations, identity-based workflows, and delegated administration.

The NetPoint Access System provides policy-based authentication, authorization, auditing, and Web single sign-on. All Access System components are involved with Smart Card authentication, as discussed next:

- **The Access Manager**—The Access Manager provides the applications for policy management, resource designation (Web and non-Web) and policy testing. Master Access Administrators define policy domains and Delegated Access Administrators define the resources to be protected by a policy domain.

When you set up the Access Manager after installation, you are asked if you want to automatically configure the Client Certificate authentication scheme. This scheme is required for Smart Card authentication. Typically, a certificate must be installed on your browser and the Web server must have SSL enabled. For Smart Card authentication, however, the certificate resides on the card.

If the Client Certificate authentication scheme was configured automatically during setup, it may be used without further modification.

- **The Access System Console**—NetPoint Administrators and Master Access Administrators use the Access System Console to define and enable authentication schemes that allow or deny access to resources.

You can use the Access System Console to verify and/or modify the Client Certificate authentication scheme. However, if the Client Certificate authentication scheme was configured automatically during setup, it may be used without further modification.

- **The Access Server**—The Access Server receives requests from a WebGate and queries authentication, authorization, and auditing rules stored in the

NetPoint directory server. The Access Server returns the authentication scheme, user credentials, and authorization to the requesting WebGate.

The Access Server installation includes the cert\_decode and credential\_mapping authentication plug-ins required with the Client Certificate scheme for Smart Card authentication.

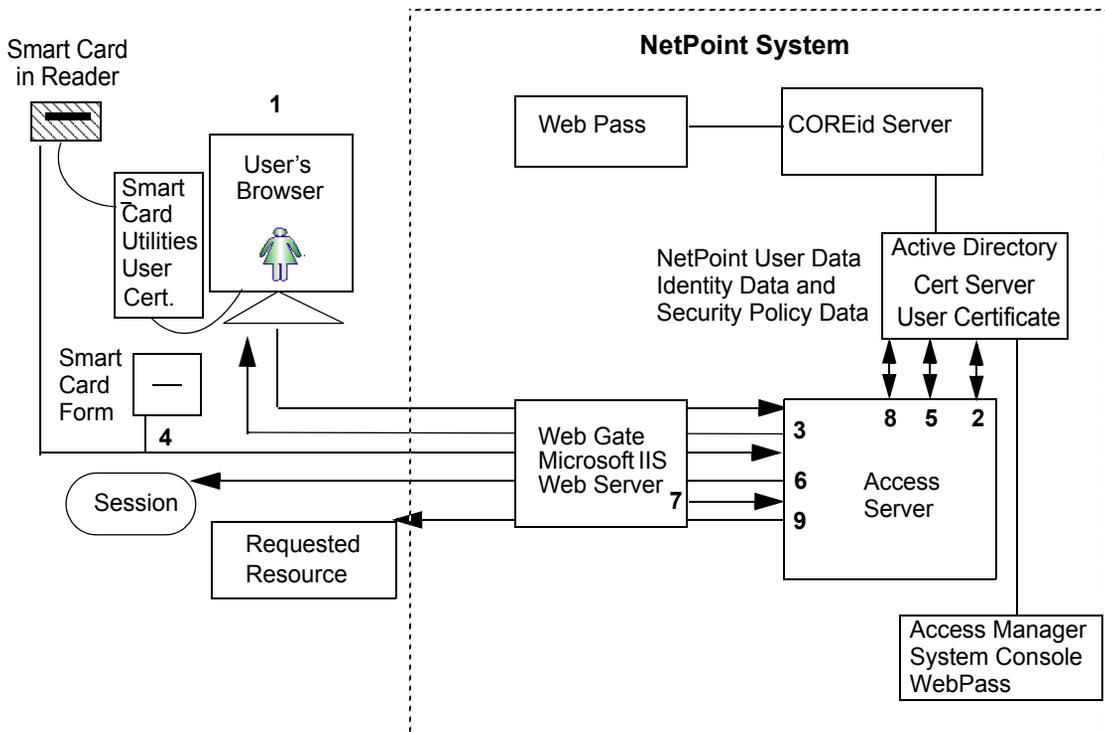
- **The WebGate(s)**—WebGates intercept and forward HTTP requests for Web resources to the Access Server for authentication and authorization. The WebGate also starts the user’s session, then creates session cookies and passes these to the user’s browser.

The cert\_authn.dll required for Smart Card authentication is provided with the WebGate installation in *WebGate\_install\_dir\access\oblix\apps\webgate\bin*. The WebGate used for Smart Card authentication must be installed with an IIS 5.0 Web server with SSL enabled.

## Integration Architecture

The following process occurs during Smart Card authentication with NetPoint. Figure 1 illustrates the sequence and is followed by a process overview.

**Figure 1** ActivCard Authentication Sequence



## **Process overview: During Smart Card authentication**

1. The browser prompts the user for the Smart Card and the WebGate intercepts the user's resource request and queries the Access Server to determine if and how the resource is protected, and if the user is authenticated.
2. The Access Server queries the Active Directory server for authentication information and receives information from the directory.
3. The Access Server responds to the WebGate, which prompts the browser to challenge the user to either insert their ActivCard and/or enter their PIN.
4. The user submits their credentials, which the browser passes to the WebGate and the WebGate presents to the Access Server, at which point one or more authentication plug-ins are used.

The cert\_decode and credential\_mapping plug-ins are required with the Client Certificate authentication scheme.

5. The Access Server performs the authentication dialog with the Active Directory, which maps the certificate information stored in the Smart Card to the user certificate in the directory and returns a success response to the Access Server.
6. When the user's credentials are valid, the Access Server provides the response to the WebGate, which starts a session for the user.
7. The WebGate queries the Access Server for resource authorization.
8. The Access Server queries Active Directory for authorization information that allows or denies access based upon the policy domain's authentication and authorization rules.
9. When access is granted, the Access Server passes authorization to the WebGate, which presents the resource to the user.

# Supported Versions and Platforms

NetPoint 7.0 supports Smart Card authentication in the environments in Table 29:

**Table 29** NetPoint Smart Card Support

Requirements
ActivCard Gold v2.2a Client, USB, or PCMCIA Card Reader
Homogeneous Windows environments with Active Directory
Windows 2000 Server SP4 and Windows XP
IIS Web server
Microsoft IE 5.5 or 6.0 SP1 browser Netscape 6.2.2

See ActivCard Gold specifications and details about Smart Card standards and compatibility at [www.activcard.com](http://www.activcard.com).

## Setting Up Smart Card Authentication

Several procedures must be completed to set up Smart Card authentication with NetPoint 7.0.

### Task overview: Setting up Smart Card Authentication

1. Confirm your environment meets requirements in “Supported Versions and Platforms” on page 485.
2. “Preparing the Active Directory” on page 485
3. “Preparing the CA and Enrolling for a Certificate” on page 486.
4. “Preparing IIS Web Servers” on page 487.
5. “Preparing NetPoint for Smart Card Authentication” on page 487
6. “Protecting Resources with NetPoint” on page 487
7. “Setting Up the IIS Manager” on page 490

## Preparing the Active Directory

For more information about this procedure, see the Active Directory manual. For details about setting up your Active Directory to operate with NetPoint, see the *NetPoint 7.0 Installation Guide* and *NetPoint 7.0 Administration Guide Volume 1*.

## To prepare the Active Directory

1. Ensure that you have a domain controller and Active Directory installed and properly running.
2. Ensure that you have a Domain Name System (DNS) server installed and properly running.

---

**Note:** You must install a Microsoft certification server with Active Directory, as discussed next.

---

## Preparing the CA and Enrolling for a Certificate

For more information about the following tasks, see the ActivCard documentation, *Configuring Smart Card logon with ActivCard CSP for Windows 2000*.

### To prepare a certification authority

1. Confirm that you have met all setup requirements for certification authorities (CAs), install ActivCard Gold utilities, and set up the CA.

If you want the user's certificate installed on the ActivCard only, rather than on both the machine and the ActivCard, you need at least two installations of the ActivCard Gold utilities because you need an administrator's certificate to digitally sign a user's certificate.

2. Establish the certificate types that an enterprise certification authority can use.
3. Prepare a certification authority to issue Smart Card certificates.

### To complete Smart Card certificate enrollment

1. Prepare a Smart Card certificate enrollment station on a computer that you will use to set up smart cards and install a ActivCard USB reader v2.0.

If you want the user's certificate installed on the ActivCard only, rather than on both the machine and the ActivCard, you need multiple ActivCard USB Readers and at least two ActivCard Gold.

2. Connect a Smart Card reader.
3. Enroll for a Smart Card Logon or Smart Card User certificate, initialize the card, and digitally sign the request.

For more information about downloading certificates onto ActivCards, see the *ActivCard Gold User Guide*.

4. Log on with an ActivCard, as described in *Configuring Smart Card logon with ActivCard CSP for Windows 2000*.
5. Set policies for Smart Card removal behavior.

## Preparing IIS Web Servers

For more information, see the Microsoft documentation for your IIS Web server.

### To prepare the IIS Web server for certification authentication

1. Deploy a certificate and the CA that issued the certificate within IIS on the Web server that hosts the WebGate.
2. Enable SSL to protect communication on port 443 on the Web server that hosts the WebGate.
3. Enable client certificate authentication within IIS.
4. Download a 1024-bit-length Web server certificate from your Microsoft certificate server.

---

**Note:** Do not use a 512-bit-length certificate.

---

## Preparing NetPoint for Smart Card Authentication

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

### To prepare NetPoint for Smart Card authentication

1. Ensure that your NetPoint system is properly installed and running with Active Directory, including the latest patches. For example:
  - a) COREid Server and WebPass
  - b) Access Manager and Access System Console
  - c) Access Server and WebGates
2. Confirm that SSL is enabled on the IIS Web server hosting the WebGate.

## Protecting Resources with NetPoint

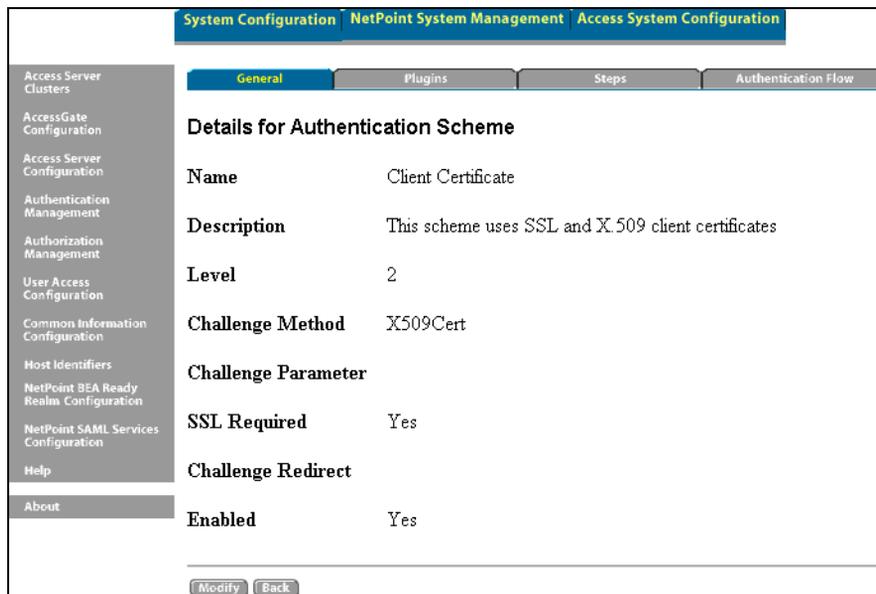
You need to modify the Client Certificate authentication scheme and add it to a policy domain to protect resources for Smart Card authentication.

Steps are provided in this procedure. For additional information, see the *NetPoint 7.0 Administration Guide Volume 2*.

### To configure the authentication scheme for Smart Card

1. Navigate to the Authentication Management page: Access System Console > Access System Configuration > Authentication Management.
2. Create or modify the Client Certificate authentication scheme to use the X509Cert challenge method, as shown in the example in Figure 2.

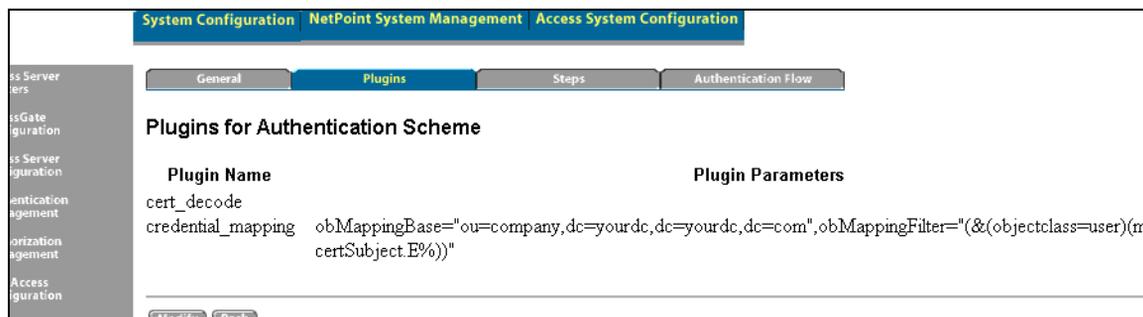
**Figure 2** Client Certificate Authentication Scheme for Smart Card



3. Click the Plug-Ins tab and ensure that the cert\_decode and credential\_mapping plug-ins contain appropriate parameters and values for Smart Card authentication, as shown in the example in Figure 3.

For more information, see “Client Certificate Authentication Schemes” on page 491.

**Figure 3** Smart Card Authentication Scheme Plug-In Parameters



This scheme will appear in the Authentication Scheme drop-down list when you add authentication rules to the policy domain.

Next, you create a policy domain in the NetPoint Access Manager. Steps are provided here and additional information is available in “About Policy Domains for Smart Card Authentication” on page 490.

## To protect resources

1. Navigate to the Access Manager, as usual:

`http://hostname:port/access/oblix`

2. Create a policy domain, as usual: Access Manager > Create Policy Domain.

For example:

**Name**—Your Choice.

**Description**—Optional

---

**Note:** Do not enable the policy domain until all specifications are completed.

---

3. Click Save.
4. Click the Resources tab, then click Add and add a resource.

For example:

**Resource Type**—Your Choice

**URL Prefix**—Your Choice

**Description**—Optional

5. Click Save.
6. Click Authorization rules, and configure those that apply to your policy domain and resource, then confirm or add plug-in parameters, as usual.
7. Click the Default Rules tab, click the Add button, enter the details for the authentication rule and confirm that you are using the modified Client Certificate authentication scheme.

For example:

**Name**—Your choice

**Description**—Optional

**Authentication Scheme**—Client Certificate

8. Add an access policy, as needed.

Delegating Administration is done as usual. There are no special requirements. For more information, see the *NetPoint 7.0 Administration Guide Volume 1*.

9. Click the General tab and enable the policy domain, as usual.
10. Continue with “Setting Up the IIS Manager” on page 490.

## Setting Up the IIS Manager

Next you must configure the NetPoint cert\_authn.dll to “accept cookies”, in the Internet Services Manager

### To configure the cert\_authn.dll

1. Navigate to the Internet Services Manager: Start > Programs > Administrative Tools > Internet Services Manager.
2. Expand the host, double click the Default Web Site (or another Web site if you are not using the default), then navigate to and double-click the cert\_authn.dll.

For example:

```
hostname > Default web site  
access\oblix\apps\webgate\bin\cert_authn.dll
```

---

**Note:** If the ISAPI WebGate installation configuration is performed manually, the information below will be presented on an HTML page.

“If you are using client certificate authentication you must enable client certificates for the WebGate and SSL must be enabled on the IIS Web server hosting the WebGate. Once this is done, do the following steps to enable client certificates for the WebGate:”

---

3. Select the File Security tab, then click Edit in the Secure Communications panel at the bottom of the window: File Security > Secure communications Edit.
4. In the Client Certificate Authentication subpanel, enable Accept Certificates.
5. Click OK in the Secure Communications window, and click OK in the cert\_authn.dll Properties window.

## About Policy Domains for Smart Card Authentication

The key to creating an effective policy domain is to group the content that you want to manage in the same way. In this case, you will group resources that require Smart Card authentication under one policy domain.

Each policy domain includes a definition of the authentication scheme, rules, optional policies, administrative rights, and resources to protect. Only one authentication rule is allowed per policy domain or policy. Only one authentication scheme is allowed per rule to enforce authentication. The default rule applies unless you set overriding policies (exceptions) for specific resources (URL patterns).

**Authentication Scheme**—An existing authentication scheme must be specified as the building block for a rule. The Client Certificate authentication scheme is required for Smart Card authentication. For more information, see “Client Certificate Authentication Schemes” on page 491.

**Administrative Rights**—Administrative rights for the policy domain are optional. Until the Master Access Administrator delegates administration rights to a policy domain, he or she is the only person who can access it. All Administrators may create an authentication rule for a policy domain or a policy (exception). Only a Master Access or NetPoint Administrator may add resources to a policy domain.

**Resources**—Resources may be either static content such as HTML pages, .gifs, and .pdfs, or dynamic content such as scripts, applications, and EJBs.

For more information about policy domains, see the *NetPoint 7.0 Administration Guide Volume 2*.

## Client Certificate Authentication Schemes

NetPoint automatically configures the default Client Certificate authentication scheme if the NetPoint Administrator selected this option during Access System installation. This scheme may be set up and/or modified after installation.

The Client Certificate scheme indicates that the user must supply a digital certificate to the policy domain to complete authentication. NetPoint supports client certificate authentication using public key encryption cryptography and X.509 certificates.

Your organization can determine how to obtain a certificate; there are no NetPoint requirements for this.

When you use the Oblix-provided schemes and plug-ins, you must be sure the `obMappingFilter` of the plug-in parameter is set correctly for your directory and environment. For additional information, see:

- “Smart Card Challenge Method, Parameter, SSL” on page 492
- “Plug-Ins for Smart Card Authentication” on page 492
- *NetPoint 7.0 Administration Guide Volume 2* for details on protecting resources using policy domains.

## Smart Card Challenge Method, Parameter, SSL

Each authentication scheme requires a challenge method to obtain user credentials for authentication. Only one challenge method is allowed per authentication scheme. Smart Card authentication has no Challenge Redirect requirement; however, the following is required:

- Smart Card authentication requires the X509Cert Challenge Method and X509 Challenge Parameter, which support public key encryption cryptography and X.509 certificates.
- Smart Card authentication requires an SSL connection.

The X509Cert challenge method uses the Secure Sockets Layer (SSL) version 3 certificate authentication protocol (SSLv3) certificate authentication protocol built into browsers and Web servers. Authenticating users with a client certificate requires the client to establish an SSL connection with a Web server that has been configured to process client certificates.

---

**Note:** Smart Card authentication has no Challenge Redirect requirement.

---

## Plug-Ins for Smart Card Authentication

Two plug-ins supplied with NetPoint are required with the Client Certificate authentication scheme for Smart Card authentication. The order of execution in the Client Certificate authentication scheme for Smart Card logon is shown below.

Authentication Scheme	Plug-Ins and Order of Execution
Client Certificate	1. cert_decode 2. credential_mapping

Each plug-in defines how information will be looked up in the directory server. A number of parameters are available depending upon the plug-in. For more information, see “cert\_decode Plug-In” on page 492 and “credential\_mapping Plug-In” on page 493.

If your certificate is stored in the browser, you can view the certificate details. For more information, including the OIDs of the attributes that are supported by the Access Server with the corresponding suffix used to retrieve the attribute, see the *NetPoint 7.0 Administration Guide Volume 2*.

### cert\_decode Plug-In

The cert\_decode plug-in can be used with the X509Cert challenge method and must be included in the Client Certificate authentication scheme for Smart Card authentication.

The `cert_decode` plug-in has no parameters and does not use a data source. This should be the first plug-in in the Client Certificate authentication scheme for Smart Card authentication.

`cert_decode` decodes the certificate and extracts the components of the certificate subject's and issuer's Distinguished Name. For each component, the plug-in inserts a credential with a `certSubject` or `certIssuer` prefix. For instance, if your certificates have a subject name such as `givenName=somename`, the plug-in will add the credential `certSubject.givenName=somename` to the credential list.

If decoding is successful, the elements of the certificate's subject and issuer DN are added to the list of credentials. If not, authentication fails.

## **credential\_mapping Plug-In**

The `credential_mapping` plug-in can be used with the `X509Cert` challenge method and must be included in the Client Certificate authentication scheme for Smart Card authentication.

The `credential_mapping` plug-in should be second in the Client Certificate authentication scheme for Smart Card authentication. This plug-in maps the user-provided information to a valid Distinguished Name (DN) in the NetPoint directory using the parameters below:

```
obMappingBase="ou=company,dc=yourdc,dc=yourdc,dc=com"  
obMappingFilter="(&(objectclass=user)(mail=%certSubject.E%))"
```

You can configure the attribute to which the `userID` is mapped to find the DN by changing the `obMappingFilter` parameter as shown above, where:

```
dc=the Active Directory Domain Controller  
mail=%certSubject.E%=maps the email in the Active Directory to the email in  
the certificate
```

With these concepts in mind, complete the steps under "Protecting Resources with NetPoint" on page 487.

# Troubleshooting

Several troubleshooting tips for Smart Card authentication are discussed below.

- “Problem Requesting X.509 Certificates” on page 494
- “Additional Resources” on page 494

## Problem Requesting X.509 Certificates

NetPoint requires X.509 certificates from Microsoft's Certification Server on Windows 2000 to be downloaded to the Smart Card. In this case, you need the ActivCard Gold for authentication.

### Problem

You request a certificate for Smart Card from the Web page below:

<http://hostname/certsrv/certsces.asp>

and see the message “Downloading ActiveX Controls...” yet never complete the process.

### Solution

1. Visit the Web page identified below.

<http://www.microsoft.com/windows2000/downloads/critical/q323172/default.asp>

2. Obtain security patch Q323172 for certificate downloads with IIS.

## Additional Resources

There are several sources of information that you may find useful when setting up Smart Card authentication for NetPoint 7.0.

### Active Directory Resources

For more information about setting up Active Directory, see:

- Microsoft Active Directory documentation
- *NetPoint 7.0 Installation Guide* chapter on installing on Active Directory
- *NetPoint 7.0 Administration Guide Volume 1* for details on deploying with Active Directory

## Smart Card Resources

For more information about setting up ActivCard utilities and the Smart Card, see the documentation that accompanies your ActivCard product packages, including:

- ActivCard Gold User Guide
- ActivCard: Configuring Smart Card logon with ActivCard CSP for Windows 2000
- ActivCard Trouble Shooting Guide

For general information about smart cards, see:

- Microsoft Step-by-Step Guide to Installing and Using a Smart Card Reader
- Microsoft Step-by-Step Guide to Mapping Certificates to User Accounts

## NetPoint Policy Domain Details

For more information about setting up protecting resources with NetPoint policy domains, see the *NetPoint 7.0 Administration Guide Volume 2*.



# 11 Integrating NetPoint with .NET Passport

Integrating NetPoint 7.0 with Microsoft .NET Passport allows NetPoint to use Passport as an external authentication provider.

This chapter explains how to integrate NetPoint with .NET Passport. It includes the following topics:

- “About NetPoint and .NET Passport” on page 497
- “Supported Platforms, Versions, and Requirements” on page 499
- “Integration Architecture” on page 500
- “Integrating NetPoint and the Passport Plug-In” on page 501

## About NetPoint and .NET Passport

The Microsoft .NET framework includes a component called .NET Passport which provides authentication services for single sign-on between Passport protected Web sites. To support .NET Passport, NetPoint provides certified integration software called the NetPoint Passport Authentication Plug-in.

The NetPoint 7.0 Passport Authentication Plug-in enables Microsoft .NET Passport to act as the authentication service while NetPoint enforces local authorization for protected resources. NetPoint also provides single sign-on between .NET Passport sites and NetPoint-protected sites. Here is how the functions are divided:

- .NET Passport acts as the authentication method only.
- NetPoint provides the authorization and identity management infrastructure.
- NetPoint FEDERATEDid Layer provides the point of integration.

The Passport authentication service holds user authentication information—such as email addresses and passwords, and user profile information such as birth dates and occupations—that users can choose to share with Web sites using Passport. Each registered user is assigned a Passport Unique ID (PUID) that is the primary user identifier.

The NetPoint Passport Authentication Plug-in can be used for many purposes that rely on single sign-on. For example, a financial services enterprise organization might use NetPoint to protect its Web applications. Assuming this case, a customer, such as a consumer banker, logs into a Microsoft Passport-enabled Internet site where he is authenticated. A PUID is generated as part of the Passport authentication process. After authentication, the user attempts to access on the NetPoint-protected financial services Web site an application which enables him to consolidate account information.

### **Process overview: When the user signs in to the Passport service**

1. Passport challenges the user for credentials.
2. If the user authenticates successfully, Passport sets several encrypted cookies with user identity and profile data.
3. When a Passport-authenticated user attempts to access a NetPoint-protected resource:
  - NetPoint extracts the PUID and the user's preferred email address from the session token stored in one of the Passport-specific cookies.
  - NetPoint maps the PUID (and other information such as the user's preferred email address) extracted from the session token to information in a NetPoint user profile in the NetPoint LDAP directory.

The Passport PUID or user's preferred email address serves as credentials for NetPoint's own user credential mapping authentication process. NetPoint does not need to prompt the user for credentials, but rather it maps the already validated credentials to corresponding information in the NetPoint LDAP directory.
  - NetPoint automatically creates a NetPoint session token for the user.
4. After the user's Passport credential (PUID or email address) is mapped to NetPoint user profile information in the NetPoint LDAP directory, the user can access resources protected by NetPoint with having to authenticate again. However, the user must first be authorized by NetPoint to use the requested resource.

# Supported Platforms, Versions, and Requirements

This discussion provides information on platforms, versions and required components.

NetPoint 7.0 supports .NET Passport for the following IIS Web servers on the platforms indicated below:

IIS Version	Platform
6.0	Windows Server 2003 <b>Note:</b> .NET Passport Authentication must be selected
5.0	Windows 2000 Server (SP4) <b>Note:</b> Passport SDK is required

## Required Passport Components

The following components are required to integrate NetPoint with Passport.

- **Passport Manager**—An ISAPI filter plug-in that is provided by Microsoft as part of the Passport Service Developers Kit (SDK).

This filter is responsible for obtaining the user identity and profile information from the Passport Authentication Service.

---

**Note:** The Passport Manager plug-in is built into IIS 6.0.

---

### Task overview: To use the Passport Plug-in

- On IIS 5.0 or Windows 2000, you must install the Passport SDK.
- On Windows 2003, you must ensure that .NET Passport Authentication is selected.

Here is the path to follow to reach the configuration screen where you can turn on the .NET Passport authentication for MS Windows 2003:

IIS Manager > Directory > Security > Properties > Authentication Methods  
.NET Passport Authentication

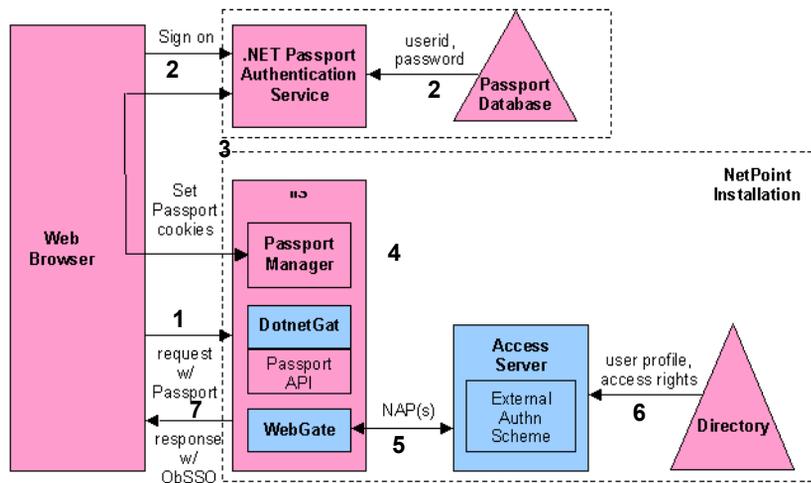
## Required NetPoint Components

The following NetPoint components are required to integrate NetPoint and Passport:

- **WebGate**—An ISAPI plug-in that intercepts HTTP requests for Web resources and forwards them to the Access Server for authentication and authorization.
- **DotNetGate**—An ISAPI plug-in that passes Passport authentication information to WebGate.
- **Access Server**—One or more stand-alone servers that provides authentication, authorization, and auditing services.
- **Access Manager**—A user interface through which you can define resources, and create and manage policies to protect them.
- **Access System Console**—A user interface through which you can create an external authentication scheme to be used for Passport authentication.

## Integration Architecture

The following diagram illustrates the integration between NetPoint and Passport. Following the diagram is a description of the interactions.



## **Process overview: NetPoint and Passport integration**

1. A user attempts to access a resource requiring Passport authentication.
2. Passport challenges the user for credentials and validates the user name and password to authenticate the user.
3. If the user authenticates successfully, the Passport Manager sets several cookies and generates a session token that contains user attributes such as the PUID and the email address.

See Microsoft .NET Passport documentation for information on cookies set by Passport Manager.

4. DotNetGate calls the Passport API, extracts the PUID and the email address from the session token stored in the Passport cookie, and passes the information onto WebGate.
5. WebGate passes the PUID and the user's preferred email address to the Access Server.
6. The Access Server maps the PUID or the preferred email address to a user profile attribute in the NetPoint LDAP directory.

The external authentication scheme configured for Passport authentication includes parameters that specify the attributes to be mapped.

You must create an external authentication scheme containing this information. See “Creating an External Authentication Scheme for Passport Authentication” on page 504 for details.

7. The Access Server creates a NetPoint session token for the mapped user.

## **Integrating NetPoint and the Passport Plug-In**

To integrate NetPoint 7.0 with the Passport Plug-in, you must complete the following tasks, which are explained in this section:

- “Preparing for the Integration” on page 502
- “Installing DotNetGate on IIS” on page 502
- “Configuring NetPoint” on page 503

## Preparing for the Integration

Before you begin the integration process, you must complete some preliminary processes.

---

**Note:** Steps 1 and 2 are not required for the Windows Server 2003.

---

### To prepare to integrate NetPoint with the Passport Plug-in

1. Install and set up the Passport Plug-in, as described in the Microsoft .NET Passport documentation.
2. Add the Passport Manager filter on IIS, as described in the Microsoft .NET Passport documentation.
3. Configure at least one Web server within your domain to use the Passport Authentication Service, as described in the Microsoft .NET Passport documentation for the Passport Plug-in.
4. Install and configure NetPoint WebGate on IIS, as described in the *NetPoint 7.0 Installation Guide* and “Installing DotNetGate on IIS” on page 502.
5. Configure NetPoint, as described in “Configuring NetPoint” on page 503.

## Installing DotNetGate on IIS

NetPoint provides a plug-in named DotNetGate that connects NetPoint to the Passport Authentication Service. DotNetGate enables NetPoint-protected Web servers to use Passport as an authentication service. DotNetGate must be installed on IIS.

### To install DotNetGate

1. On the IIS WebGate, navigate to Start > Programs > Admin Tools > Internet Information Server.
2. Using the IIS Console, navigate to *localComputer*/Web Sites, where *localComputer* is your local system.
3. Right-click Web Site and select Properties.
4. Select the ISAPI Filters tab.
5. Click Add.

The Filter Properties box appears.

6. Enter DotNetGate in the text field
7. Click Browse and navigate to the location of the dotnetgate.dll file.
8. Click Open.

DotNetGate is appended to the list of filters displayed in the ISAPI Filters tab.

9. Click Apply.
10. Move DotNetGate between Passport Manager and WebGate using the Up Arrow button in the ISAPI Filters tab.
11. Click OK to install DotNetGate on IIS.
12. Restart the Web server.

## Configuring NetPoint

You need to configure NetPoint to protect resources that require Passport authentication and NetPoint authorization, as described in the next procedures.

### **Task overview: Configuring NetPoint for Passport authentication**

1. “Setting Up a User Profile Attribute for Passport Mapping” on page 503
2. “Creating an External Authentication Scheme for Passport Authentication” on page 504
3. “Creating a Policy Domain for the Passport Authentication Scheme” on page 505

## Setting Up a User Profile Attribute for Passport Mapping

Before you begin to set up the NetPoint Access System to support the NetPoint Passport Authentication Plug-in feature, set up a user profile attribute to map the Passport credential to.

### **To set up a user profile attribute to map the Passport credential to**

1. Select one of the Passport credentials to be used as the mapping credential.
2. Configure a user profile attribute to map the credential to.

Oblix recommends that you use the PUID, which is always a unique value. However, you can use the user’s preferred email address.

For information describing how to configure user profile attributes, see the *NetPoint 7.0 Administration Guide Volume 1*.

# Creating an External Authentication Scheme for Passport Authentication

External authentication schemes define and refer to authentication processes other than NetPoint to be used to authenticate users. As such, they include parameters that specify user credentials passed from the external authentication service. These parameters are to be mapped to user profile attributes in the NetPoint LDAP directory. The external authentication scheme also specifies a credential mapping plug-in to be used to map the credentials.

As a Master Access Administrator, you must create an external authentication scheme in NetPoint to enable mapping a Passport user's information to equivalent NetPoint information.

## To create an external authentication scheme

1. Launch the Access System Console.
2. Navigate to Access System Configuration > Authentication Management.
3. Click Add to create an authentication scheme.
4. Enter a name and description for the authentication scheme.
5. Set the challenge method to “ext” to specify external authentication.
6. Set the challenge parameter for the user credentials that you want to map.
  - To map PUID, specify NP\_PASSPORT\_PUID.
  - To map the preferred user email address, specify NP\_PASSPORT\_EMAIL.
7. In the Plugins page, add a credential mapping plug-in that maps an attribute in the NetPoint user profile to the user credentials that you specified in the Challenge Parameter field.
8. Enable the authentication scheme.

---

**Note:** DotNetGate does not redirect users to the Passport site for authentication. It assumes that the users have already been authenticated by Passport. If the user tries to access a resource protected by the Passport authentication scheme before they are authenticated by Passport, a NetPoint mapping error occurs.

**For IIS v6**—If Passport authentication has been configured, users are automatically redirected to the Passport Website for authentication.

**For IIS v5**—Users must go to a page on a Web server that initiates Passport authentication. They are not automatically redirected.

---

For additional information, see the *NetPoint 7.0 Administration Guide Volume 2*.

## Creating a Policy Domain for the Passport Authentication Scheme

A Master Access Administrator or a Delegated Access Administrator creates a NetPoint policy domain for the resources that require Passport authentication.

You follow the normal process to create a policy domain for resources whose authentication protection process is performed by an external service, such as .NET Passport. The only difference between creating a policy domain for resources that use NetPoint authentication and a policy domain for resources that use .NET Passport authentication is specification of the external authentication scheme.

For details explaining how to create a policy domain, its policies, if any, and its authentication, authorization, and auditing rules, and authorization expressions, see the following topics in the *NetPoint 7.0 Administration Guide Volume 2*:

- Creating policy domains
- Configuring user authentication
- Configuring user authorization



# 12 Integrating NetPoint with Authorization Manager Services

NetPoint 7.0 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.

This chapter explains how to configure a NetPoint policy domain for the NetPoint 7.0 AzMan Plug-in, and includes the following discussions:

- “About NetPoint and the AzMan Plug-In” on page 507
- “Authorization with the NetPoint AzMan Plug-In” on page 509
- “NetPoint Components and Requirements” on page 512
- “About the Authorization Manager” on page 516
- “Examples” on page 521
- “Configuring the NetPoint AzMan Plug-In” on page 531
- “Troubleshooting” on page 538

## About NetPoint and the AzMan Plug-In

Authorization is the process that determines what access a user is permitted to have, and what a user is permitted to do, after they have been authenticated. NetPoint extends its access policies through authorization plug-ins.

A NetPoint authorization plug-in is a component that consists of a set of functions that reside in a dynamically-loaded native library to change or enhance NetPoint behavior. The NetPoint AzMan Plug-in allows NetPoint authorization rules to use the facilities of the Microsoft Authorization Manager on Windows Server 2003.

When using the NetPoint AzMan Plug-in:

- WebGates can control access to Web content based on Authorization Manager policies.

- Applications using the NetPoint Access Server API can use Authorization Manager policies through the `ObUserSession` `isAuthorized()` call.
- WebGates and Access Server API clients can be on any NetPoint-supported platform, which means that:
  - a) If your environment is primarily Microsoft, you can use the Authorization Manager to define policy for Windows-based applications and NetPoint can enforce those policies in the parts of the applications protected by NetPoint such as ASP URLs, for instance.

In this case, you can define application roles in the Authorization Manager and NetPoint can use these roles when enforcing Web access control.

- b) If your environment includes non-Windows applications, these applications can also use Authorization Manager policies.

Non-Windows applications can use the NetPoint Access Server SDK for the NetPoint Authorization Plug-in API to get authorization decisions from the Authorization Manager through the NetPoint AzMan Plug-in.

The NetPoint AzMan Plug-in is executed by a NetPoint Access Server during the evaluation of access policies. The NetPoint Authorization Plug-in API allows the Access Server to use the plug-in to make outbound calls to external business logic. The external business logic determines whether a user is authorized to access a resource. The external business logic also determines whether to pass authorization actions during the evaluation of access policies.

The NetPoint Authorization Plug-in API:

- Provides the interface that the NetPoint AzMan Plug-in implements and the Access Server calls
- Provides the callback functions that the plug-in uses to retrieve additional data from the Access Server
- Defines data structures that pass information between the NetPoint AzMan Plug-in and the Access Server

For example, the `ObUserSession.isAuthorized()` method in the Access Server SDK for the NetPoint Authorization Plug-in API can evaluate AzMan policies for a user and, optionally, for a set of parameters.

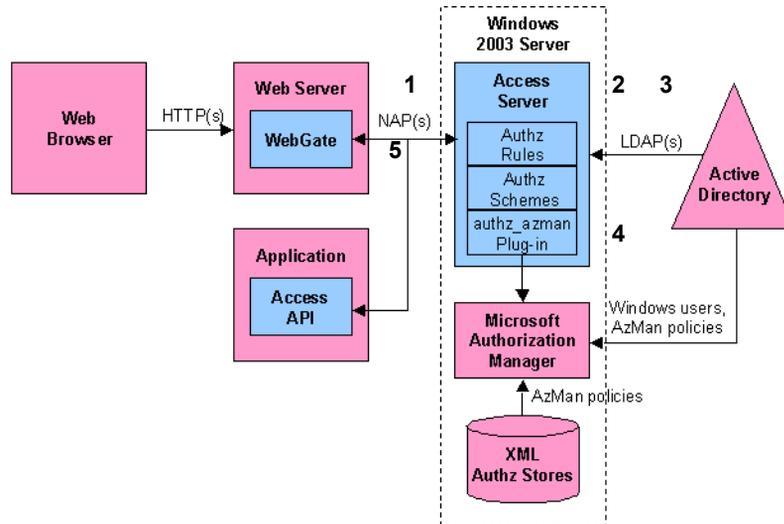
For more information, see the following discussions:

- “Authorization with the NetPoint AzMan Plug-In” on page 509
- “NetPoint Authorization Rules and Schemes” on page 512
- “NetPoint Components and Requirements” on page 512

For additional information, see “About the Authorization Manager” on page 516.

# Authorization with the NetPoint AzMan Plug-In

The figure below introduces authorization with the NetPoint AzMan Plug-in.



## Process overview: WebGate operation with the NetPoint AzMan Plug-in

1. The WebGate sends an `IsAuthorized()` request for the authenticated user and the URL to the Access Server.
2. The Access Server determines the URL is protected by a policy with an authorization rule that specifies an authorization scheme for the NetPoint AzMan Plug-in.

If the authorization scheme requires request context values (configured as `RA_user` parameters in the authorization scheme) that are not available in the `IsAuthorized` request, the Access Server returns a `NeedMoreData` response to the WebGate.

When the WebGate receives the `NeedMoreData` status, the WebGate:

- a) Gets the request context data indicated in the status
  - b) Resends the `IsAuthorized` request with the data
  - c) Continues processing with the beginning of step 2.
3. If this is the first time the plug-in has been invoked, the Access Server loads the `authz_azman` library and executes the `ObAzPluginInit()` function in the library, which:

- a) Creates the authz\_azman\_log.txt file in the install\_dir/oblix/engine directory
  - b) Reads the authz\_azman\_msg.lst file in the default language directory of the installation
  - c) Checks that the Authorization API version of the Access Server is compatible with the plug-in
  - d) Initializes the COM interface
  - e) Creates a mutex to protect a global list of open application stores
4. The Access Server executes the ObAzPluginFn() function in the plug-in, which:
- a) Gets its configuration parameters from the various plug-in data blocks
  - b) Searches the list of open application stores for a store matching the AzStore parameter
  - c) If no open store is found, the plug-in opens the store and puts it in the list
  - d) Creates an application object for the AzApplication parameter
  - e) Initializes a client context for the browser user
  - f) If an AzRole is specified, the plug-in sets the client context to the role
  - g) Converts the AzOperation parameter values to an array of IDs
  - h) If the AzRuleParameters is specified, the plug-in retrieves the corresponding parameter values from the plug-in data blocks and sets up arrays with the parameters and their values
  - i) Calls the AzMan AccessCheck() method for the client context, the scope (if specified), the operation ID array, and the rule parameter and value arrays (if present)
  - j) Interprets the result of the access check:
    - If the result is access allowed and AzContinueOnAllow=yes, the plug-in returns ObAzPluginStatusContinue, which instructs the Access Server to continue processing subsequent authorization rules (possibly invoking other plug-ins).
    - If the result is access allowed and AzContinueOnAllow=no, or is omitted, the plug-in returns ObAzPluginStatusAccessAllowed this causes the Access Server to immediately return allowed.
    - If the result is access denied, return ObAzPluginStatusAccessDenied, this causes the Access Server to immediately return denied.

5. The WebGate gets the IsAuthorized result from the Access Server and blocks or allows access to the requested URL.

---

**Note:** For more information, see “Example 3: NetPoint AzMan Plug-In Authorization Process Flow” on page 529.

---

### **Process overview: Access Server API operation with the NetPoint AzMan Plug-In**

An application using the Access Server API executes the ObUserSession IsAuthorized() method for an authenticated user, a resource, and optional set of parameters that may include Authorization Manager configuration parameters. The following process overview the operation of the Access Server API and the NetPoint Authorization Manager Plug-in.

1. The Access Server API sends an IsAuthorized request with the authenticated user, resource, and parameters to the Access Server.
2. The Access Server determines the resource is protected by a policy with an authorization rule that specifies an authorization scheme for the AzMan plug-in.
3. If the authorization scheme requires request context values (configured as RA\_user parameters) that are not available in the IsAuthorized request, the Access Server returns a NeedMoreData status to the Access Server API.

---

**Note:** This is not as likely to happen as with WebGate, since the application can include the required parameters in the isAuthorized() call.

---

4. If Access Server API gets the NeedMoreData status, it gets the request context data indicated in the status from the resource (for example, a query string) and resends the IsAuthorized request with the data.
5. Processing then continues with step 2 in “Process overview: WebGate operation with the NetPoint AzMan Plug-in” on page 509.
6. Steps 3 and 4 in this process are the same as in “Process overview: WebGate operation with the NetPoint AzMan Plug-in” on page 509. The parameters from the isAuthorized() call are in the request context data block.
7. The Access Server API client in the application gets the IsAuthorized result from the Access Server and returns the result through the isAuthorized() call. The application then takes appropriate action.

For more information, see “Using the NetPoint AzMan Plug-In with the Access Server API” on page 536.

# NetPoint Components and Requirements

The plug-in is included during NetPoint Access Server installation in the following library:

```
AccessServer_install_dir\access\oblix\lib\authz_azman.dll
```

The NetPoint AzMan Plug-in must be installed on each application server you want to protect. This enables you to complete authorization for resources protected by NetPoint using policies and roles defined outside the NetPoint policies, within the Authorization Manager in Active Directory, through the NetPoint Authorization Plug-in API. For more information about the NetPoint Authorization Plug-in API, see *NetPoint 7.0 Developer Guide*.

---

**Note:** NetPoint does not provide or allow administration of Authorization Manager policies through the NetPoint Access Manager.

---

NetPoint provides the custom AzMan Plug-in but not a custom authorization scheme, because external programs and calls to external business logic are unique from business to business.

A Master Access Administrator uses the NetPoint Access System Console to define a custom authorization scheme that includes the full path to the shared library for the NetPoint AzMan Plug-in.

A Delegated Access Administrator uses the NetPoint Access Manager to define a NetPoint policy domain using the custom authorization scheme and plug-in parameters for the NetPoint AzMan Plug-in as a basis for the authorization rules or action in a policy domain, and to protect resources.

For more information, see “Configuring the NetPoint AzMan Plug-In” on page 531.

## NetPoint Authorization Rules and Schemes

This discussion explains NetPoint authorization rules and schemes and NetPoint AzMan Plug-in parameters for use with the Authorization Manager.

A NetPoint authorization rule allows or denies users the right to access the resources within the policy domain (or a subset of resources, if a policy applies). Authorization rules can be combined into expressions. For more information about chaining authorization rules, see the *NetPoint 7.0 Administration Guide Volume 2*.

**Table 30** You define and enable authorization rules through the Access Manager. When using the NetPoint AzMan Plug-in, you need to create a custom authorization scheme that consists of a name, a description, a shared library path for the installed plug-in (without a platform-specific extension such as .dll), and a set of required and optional parameters. The purpose of the authorization

scheme parameters is shown in Table 31.

**Table 31** Purpose of Authorization Scheme Parameters

Scheme Parameters	Description
User Parameters <ul style="list-style-type: none"><li>• User</li></ul>	User profile attribute values passed into the plug-in in the RequesterInfo data structure. For more information, see Table 33 on page 514.
<ul style="list-style-type: none"><li>• Request Context</li></ul>	Request data (HTTP headers and cookies, Access Server API parameters) passed into the plug-in in the RequestContext data structure. <ul style="list-style-type: none"><li>• Introduced in NP 6.1.1 and defined as user parameters with the prefix RA_. If not available in the request, the access check will fail.</li><li>• For more information, see Table 33 on page 514.</li></ul>
Required Parameters	Name-value pairs passed into the plug-in in the Context data structure. These must be specified in either the authorization scheme or the authorization rule.
Optional Parameters	Name-value pairs passed into the plug-in in the Context data structure. These may be specified in either the authorization scheme or the authorization rule or may be omitted.

The NetPoint AzMan Plug-in uses optional plug-in parameters to specify the input to the `AccessCheck()` method, `IAzClientContext::AccessCheck()`, discussed in “Using the NetPoint AzMan Plug-In with the Access Server API” on page 536. If a plug-in parameter is not specified, the plug-in will check the User Parameters, Request Context data (see Table 31), for the omitted values. In this case, callers of the Access Server API can supply these parameters in the `ObResourceRequest` constructor or the `ObUserSession` `isAuthorized()` call. `AccessCheck()` can return a value indicating that access is allowed or denied. The plug-in can take a different action based on the `AzContinueOnAllow` configuration parameter in Table 32. For details about the Access Server API, see the *NetPoint 7.0 Developer Guide*.

The plug-in parameters shown in Table 32 are specific to the NetPoint AzMan Plug-in. You use them to specify input to the Authorization Manager.

**Table 32** NetPoint AzMan Plug-in Parameters

Parameters	Description
AzStore	URL (msldap:// or msxml://) identifying the authorization store with the relevant policies.
AzApplication	Name of the application in the store containing the policies to be used. This must be specified.
AzObject	Name of the object to be identified in the AzMan audit log. If not specified, the NetPoint resource URL will be used.
AzScope	Name of the scope in the application containing the policies to be used. If not specified, no scope will be used and the default application policies will be applied.
AzOperations	Space-separated list of operation names to be used in the access check. Operation names with embedded spaces must be enclosed in quotation marks such as "an operation". If not specified, the NetPoint resource request operation name will be used.
AzRuleParameters	Space-separated list of names of parameters to be passed to AzMan authorization rules. Parameter names with embedded spaces must be enclosed in quotation marks such as "a parameter name".
AzContinueOnAllow	<ul style="list-style-type: none"> <li>• If AzContinueOnAllow=yes, the plug-in will return a continue status to the Access Server, executes subsequent authorization plug-ins, if any.</li> <li>• If AzContinueOnAllow=no, or is omitted (the default), the plug-in will return an allow status and the Access Server will immediately return an allowed status for the policy evaluation.</li> </ul>
AzLogLevel	<ul style="list-style-type: none"> <li>• If high, all authorization requests with their parameters and result (allow, deny, continue) will be logged.</li> <li>• Otherwise only errors are logged in: <i>AccessServer_install_dir\oblix\engine\authz_azman_log.txt</i></li> </ul>

The rule parameters (specified with AzRuleParameters in Table 32) are values from either the user's profile (User Parameters) or the values from the request (Request Context parameters). Rule parameters are passed to the Authorization Manager for possible use within authorization rules/scripts.

Table 33 shows the user parameters for IAzClientContext.

**Table 33** User Parameters

User Parameters	Description
samacctuser	Username to construct the IAzClientContext object.

Table 34 shows the request context parameters that are configured as RA\_user parameters in the authorization scheme.

**Table 34** Request Context Parameters

Request Context Parameters	Description
AzRole	Value is used as the role in the access check.
<i>rule parameters</i>	Post data, query data, and all other types of data appropriate for context-specific requests can be used in an authorization decision. For post data, postgate.dll must be installed. See the <i>NetPoint 7.0 Installation Guide</i> for details.

In summary, Table 35 indicates what occurs when the Access Server evaluates a policy or policy domain that contains an authorization rule with a custom authorization scheme.

**Table 35** Summary of Evaluation

The Access Server	The Plug-In
Executes the plug-in	Extracts the parameter values from the passed data
Collects relevant parameter values for the plug-in and the target user, resource, and request.	Performs its designed tasks
Adds these values to the appropriate data structures and executes the main plug-in function	Returns a result with optional actions to the Access Server, which may include continue, allow, deny, or abort.
The Access Server interprets the result and either continues processing authorization rules or stops and returns its result to the access client.	

For more information about NetPoint authorization, see the *NetPoint 7.0 Administration Guide Volume 2*. For details about the NetPoint Authorization API, see the *NetPoint 7.0 Developer Guide*.

# About the Authorization Manager

The Windows Server 2003 Authorization Manager is a role-based access control interface characterized by using collections of settings based on an object's role within an organization. The Authorization Manager provides a GUI tool to define access policy for applications and an API for applications to request access decisions using the policy. You can use role-based administration to manage users, computers, and other file-system and directory-service objects.

The Authorization Manager provides two modes of operation:

- **Developer Mode**—Allows you to create, deploy, and maintain applications with unrestricted access to all Authorization Manager features.

You run Authorization Manager in developer mode only until the authorization store is created and configured. After you initially set up an application in developer mode, you can work in administrator mode.

- **Administrator Mode**—The default mode, allows you to deploy and maintain applications and have access to all Authorization Manager features. However, you cannot create new applications or define operations.

Before you can use administrator mode, you must provide an application that supports roles, includes all of the necessary operation and task definitions, includes its own authorization store, and is ready for use in the Authorization Manager.

## Authorization Stores

An authorization policy store contains information about the security policy of an application or group of applications. The information includes the applications, operations, tasks, users, and groups of users associated with the store.

The authorization policy store must be located on a trusted system to afford administrators on that system access to the store. The Authorization Manager supports storing authorization policy either in the Active Directory directory service or in an XML file:

- Active Directory objects are identified by an LDAP DN in a URL.

For example:

msldap:// (e.g. msldap://CN=MyAzStore, CN=Program Data, DC=authmanager, DC=com)

**or**

- XML files are identified by a path in a URL.

For example:

```
msxml://C:\MyStore.xml
```

---

**Note:** Active Directory stores allow the delegation of administrative control. However, XML stores do not.

---

By default, the group “Domain Admins” is listed within the Security tab when you create the Active Directory authorization store. To run the Authorization Manager policy through NetPoint, the Access Server user (for example, Administrator) should also be listed in the Users and Groups list within the Security tab. However, similar settings are not required for the XML store.

For more information about Authorization Stores, see your Microsoft documentation.

## Applications and Scopes

An application is a program that is designed to perform specific functions directly for the user or for another application.

An authorization store can contain policies for resources for multiple applications. Alternatively, an application’s resources and associated policies may be subdivided into scopes. For example, if you do not want to apply Authorization Manager groups, role assignments, role definitions, or task definitions to an entire application, you can create them at the scope level.

A scope can be one of the following:

- Folder
- Active Directory container
- File-masked collection of files, for example \*.doc
- URL
- Any grouping of resources meaningful to the application

You can use scopes in Active Directory authorization stores to delegate control. For more information, see your Microsoft documentation.

## Operations and Tasks

In the Authorization Manager, an operation is a small computer-level action or method of an application. Operations are grouped together as tasks. An operation is defined by:

- Name
- Description
- Operation number

---

**Note:** Operations can be defined at the application level but not the scope or store levels.

---

A task is a high-level action that users of an application need to complete. Tasks are composed of the lower-level operations required to perform the task. Users of an application request permission to complete tasks. A task is defined by:

- Name
- Description
- Set of other tasks and operations
- Authorization rule (optional)

For more information, see your Microsoft documentation.

## Roles

A role is a set of permissions that a user must have to perform the application's tasks. A role is defined by a:

- Name
- Description
- Set of tasks, operations, and other roles that are granted by the role
- Authorization rules that can test arbitrary conditions

Permissions are assigned or denied by the object's owner. The Authorization Manager is capable of implementing multiple configuration and permission changes at once and provides advantages over other management tools, such as the access control list (ACL) and Delegation of Control Wizard.

Authorization roles are based on a user's job function. You can use authorization roles to authorize access, delegate administrative privileges, or manage interaction with computer-based resources.

The Authorization Manager enables administrators to implement this role-based administration through applications. Applications using this role-based access are constructed to use logical roles that relate to the tasks performed by the application. The settings that authorize users for specific roles are made automatically through the use of scripts, called authorization rules, that enable you to control the mapping between access control and the structure of your organization.

For more information, see your Microsoft documentation.

## Groups

A group defines a set of principals (users and computers) to which roles can be assigned. A group can be defined using:

- Windows users and groups
- LDAP queries
- Other groups

A group specifies principals that are either:

- Explicitly included (members)
- Explicitly excluded (non-members)

---

**Note:** Circular group membership, for example, group A contains group B and group B contains group A, is detected and prohibited.

---

Groups can be defined at the store, application, and scope levels. Assigning a group to a role grants the role's permissions to the users defined in the group. A role definition can also contain authorization rules that can test arbitrary conditions.

For more information, see your Microsoft documentation.

## Rules

In the Authorization Manager, authorization rules are either VBScript or JScript scripts that can be used in role and task definitions. An authorization rule can determine whether the role or task is allowed. With authorization rules, you can base authorization decisions on any conditions that a script can test, including privileges and permissions, time of day, billable expense limits, account balances, and other criteria.

A rule associated with an object can regulate which users gain access and in what manner. Named parameter values can be passed from the application to the Authorization Manager for use within the scripts.

You can write your scripts in a text editor (for example, Notepad), in an integrated development environment like Visual Studio .NET, or in another application of your choice.

For more information, see your Microsoft documentation.

## Auditing

The Authorization Manager provides runtime auditing that records application-access checks using policies in an authorization store. The runtime audit log contains the relevant client contexts with the access checks. The Authorization Manager also provides authorization store-change auditing to record modifications to policies in authorization stores.

Runtime auditing can be applied at the authorization store and application levels for all stores, and at the scope level for Active Directory stores. Store-change auditing can be applied at the store, application, and scope levels for Active Directory stores, but only to the store level for XML stores.

For more information, see your Microsoft documentation.

## Authorization Manager (AzMan) API

An application program interface (API) includes the formal requests and means of communicating with other programs used by an application program. Windows Server 2003 provides Component Object Model Component Services (COM+) interfaces to manage and use Authorization Manager policies.

COM+ provides an infrastructure that enables clients and objects to work together. This binary standard enables interoperability between software components in a networked environment regardless of the language in which they were developed.

A COM client (software that uses and controls objects) does not know the internal workings of the objects (software that knows how to perform a specific task) the client is using. Clients and objects must communicate about and agree on the functionality that an object will supply to the client. This agreement is implemented in software by a COM interface.

For example, in the Authorization Manager API:

- The state of a particular user (client) is represented by an `IAzClientContext` interface.
- The object is created from one of the following:
  - a) `IAzApplication::InitializeClientContextFromToken()`: needs the user's token.

- b) `IAzApplication::InitializeClientContextFromName()`: needs the user name and domain name.
- c) `IAzApplication::InitializeClientContextFromStringSid()`: needs the string representation of the SID.
- The `AccessCheck` method of the `IAzClientContext` interface method invokes the Authorization Manager to determine if the user represented by the `IAzClientContext` object is allowed to perform a specified application operation.

For more information, see your Microsoft documentation.

## Examples

Topics here walk you through an application created with the Authorization Manager and the NetPoint configuration details for the NetPoint AzMan Plug-in.

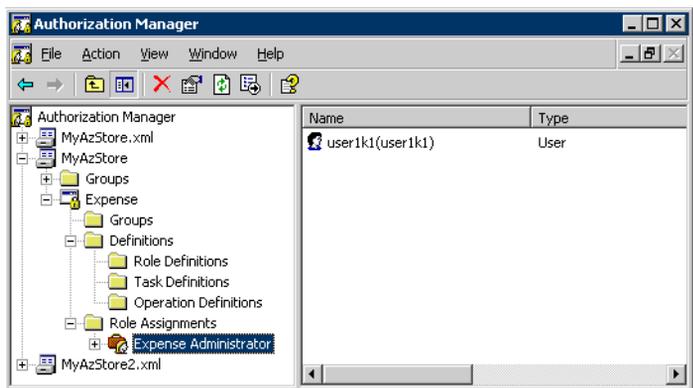
- “Example 1: An Expense Application” on page 521
- “Example 2: NetPoint Configuration” on page 525
- “Example 3: NetPoint AzMan Plug-In Authorization Process Flow” on page 529

### Example 1: An Expense Application

In this example, a financial role is defined that includes the right to authorize expenditures and audit account transactions. The Authorization Manager enables you to implement this type of role-based administration through an application that you create.

You set up the authorization store, design your application using the Authorization Manager, define tasks, operations, roles, and make role assignments. Figure 4 shows the main Authorization Manager window and the hierarchy of the authorization store, `MyAzStore`.

**Figure 4** Authorization Store Hierarchy in the Authorization Manager



In Figure 4, you can see the folders for Groups, Definitions, and Role Assignments for the application. Beneath the Definitions folder are the Role, Task, and Operation Definitions folders. In the right-hand panel, you can see the user assigned to the Expense Administrator role, user1k1.

The financial application, named “Expense”, may have the operations in Table 36.

**Table 36** Expense Application Operations

RetrieveForm
EnqueRequest
DequeRequest
UseFormControl
MarkFormApproved
SendApprovalNotify

The Expense application may include a task, “Submit Expense”, which consists of the operations in Table 36 and another task, Approve Expense, as shown in Table 37.

**Table 37** Expense Application Submit Expense Task Definition

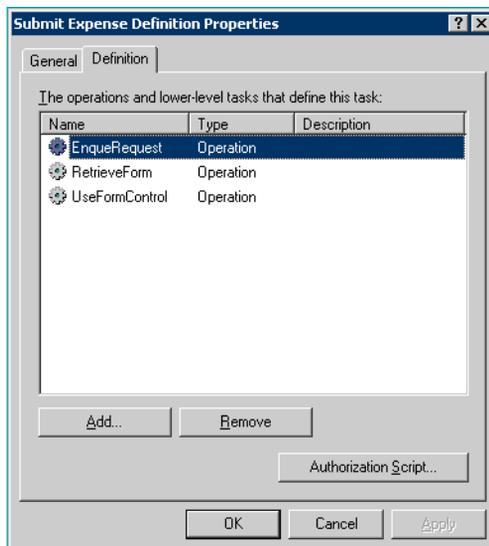
---

RetrieveForm
EnqueueRequest
UseFormControl
Approve Expense

---

Figure 5 shows the Submit Expense task, as it appears in the Authorization Manager.

**Figure 5** Submit Expense task in the Authorization Manager



The Expense application includes a role, Expense Administrator, that consists of the tasks in Table 38. A user who is assigned the Expense Administrator role is authorized to perform the operations (Table 37) to complete the Submit Expense task, among others identified below.

**Table 38** Tasks for the Expense Administrator Role

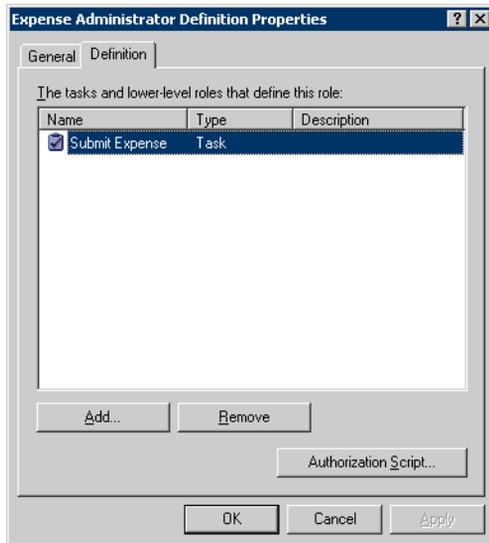
---

Submit Expense
Approve Expense
Nested role Expense Admin

---

Figure 6 shows the Expense Administrator role-definition properties in the Authorization Manager. The Submit Expense task is identified; other tasks will be added.

**Figure 6** Expense Administrator Role-Definition Properties



The Expense application may include a group of “Approvers”, to which the Expense Administrator role can be assigned. Members of the Approvers group are given permission to perform the tasks in Table 39.

**Table 39** Approvers Group Tasks

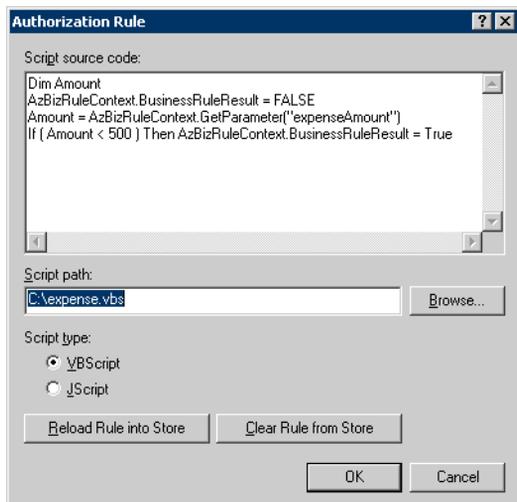
---

Submit Expense
Approve Expense tasks
Any tasks assigned to the nested Expense Admin role

---

An authorization rule for the Expense application is a script that tests the user’s expense amount (a parameter from the application) against the user’s expense limit, which could either be another application parameter or could be determined by the script itself. The rule for this Expense application is shown in Figure 7.

**Figure 7** Authorization Rule for the Expense Application



## Example 2: NetPoint Configuration

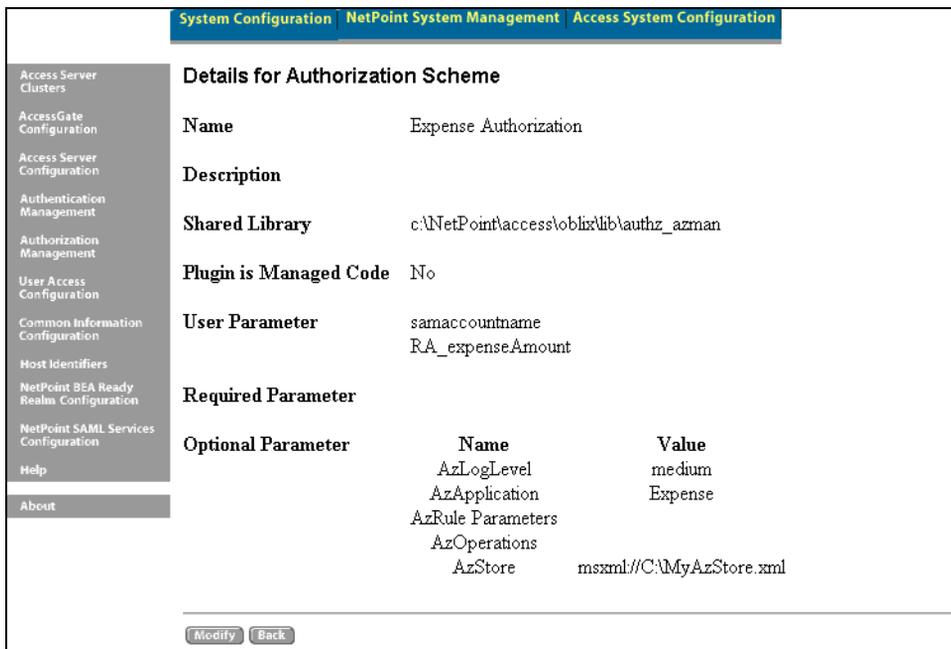
Continuing with the Expense application example described and shown in “Example 1: An Expense Application” on page 521, the following information explores the NetPoint policy domain for this application. Included is a custom authorization scheme for the NetPoint AzMan Plug-in.

The Expense application has been implemented to use Web forms to input the expense data. An XML file is used, rather than storing Authorization Manager policies in the Active Directory. Both methods are valid.

### Authorization Scheme

Figure 8 shows a custom NetPoint authorization scheme for the Expense application. Not all of the allowable NetPoint AzMan Plug-in parameters are used. Your scheme may be different.

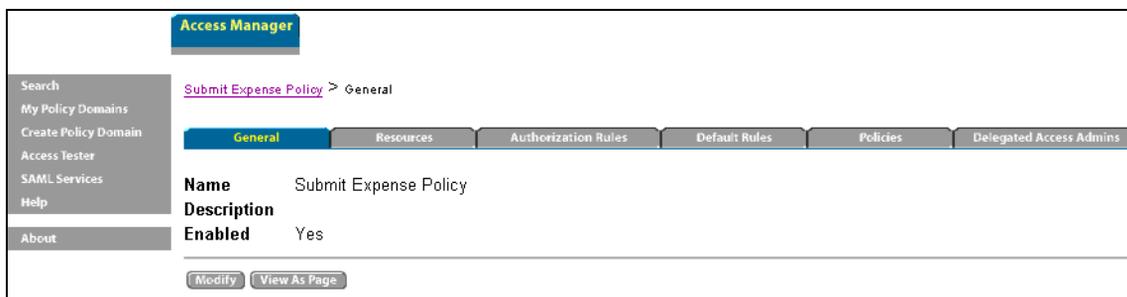
**Figure 8** NetPoint Authorization Scheme



## Policy Domain

Figure 9 shows the Submit Expense policy domain, enabled in the Access Manager.

**Figure 9** Submit Expense Policy Domain in the Access Manager



## Resources

Within the policy domain, resources have been added and protected, as shown in Figure 10 for /expense/submit.asp.

**Figure 10** Resource Types in the NetPoint Policy Domain



## Authorization Rules

The policy domain authorization rule is shown in Figure 11. Notice that it specifies the custom authorization scheme defined in the Access System Console earlier.

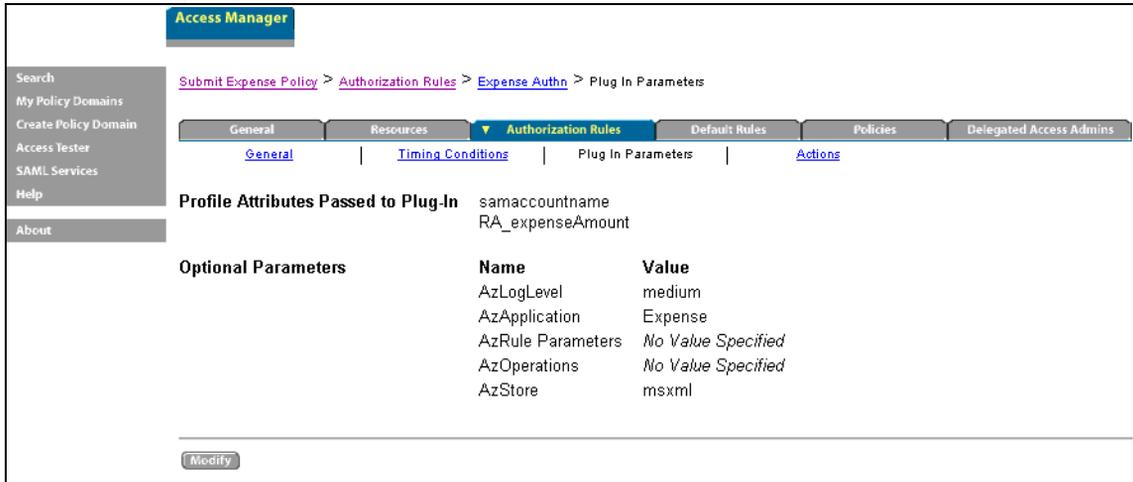
**Figure 11** Authorization Rule for the Policy Domain



There are no timing conditions for this specific authorization rule, though your rule may include these.

The plug-in parameters for this policy domain are shown in Figure 12.

**Figure 12** Policy Domain Plug-In Parameters



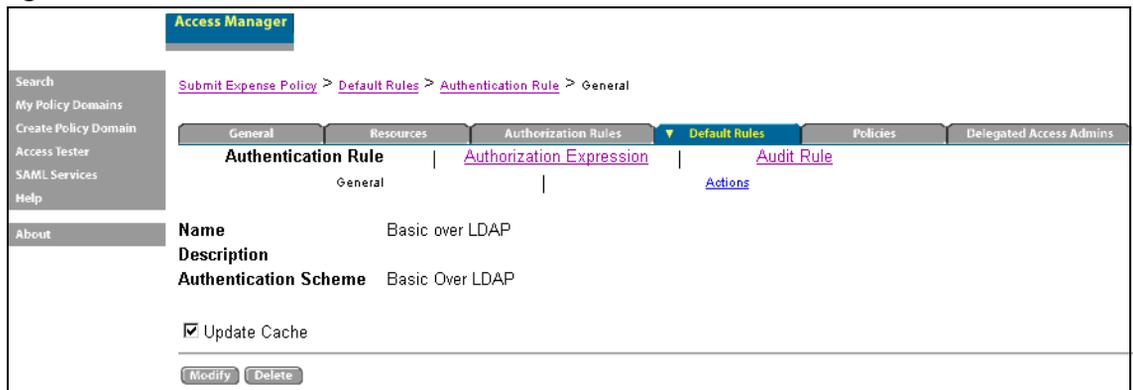
The authorization rule uses the custom Expense Authorization scheme and passes the User Parameters and AzStore and AzApplication parameters as specified in the scheme. The rule adds an AzOperations value for the EnqueRequest operation and an AzRuleParameters value for the expenseAmount variable.

There are no actions associated with this particular rule; however, your application may have specific actions.

## Default Rules

The default authentication rule for the policy domain is shown in Figure 13. There are no authorization expressions or audit rules for this policy domain. Your environment may be different.

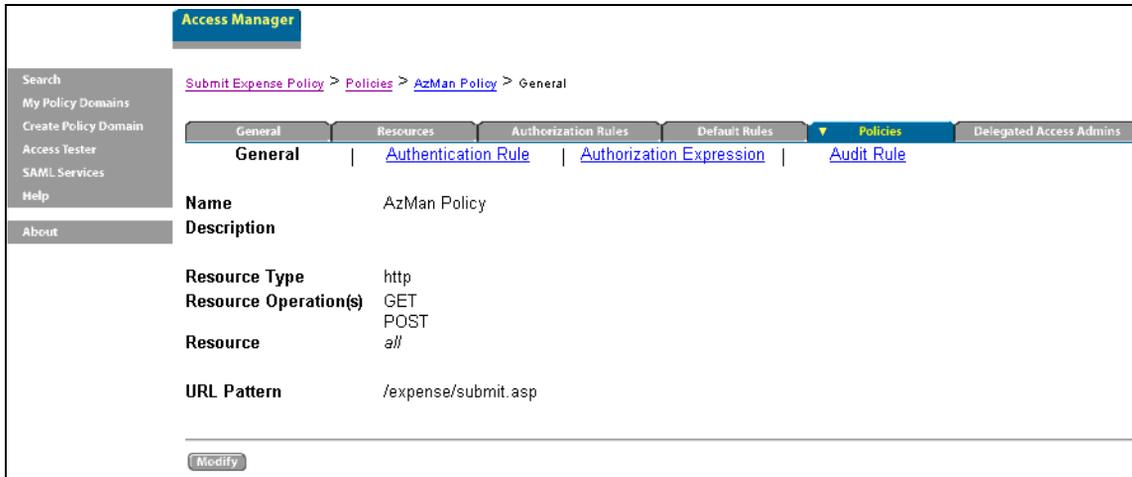
**Figure 13** Default Authentication Rule



# Access Policy

Figure 14 shows the access policy for /expense/submit.asp.

**Figure 14** Access Policy for /expense/submit.asp



There are no authentication rules, authorization expressions, or audit rules defined for this policy.

## Delegated Access Administrators

Delegated Access Administrators are defined for this policy domain, but are not shown here.

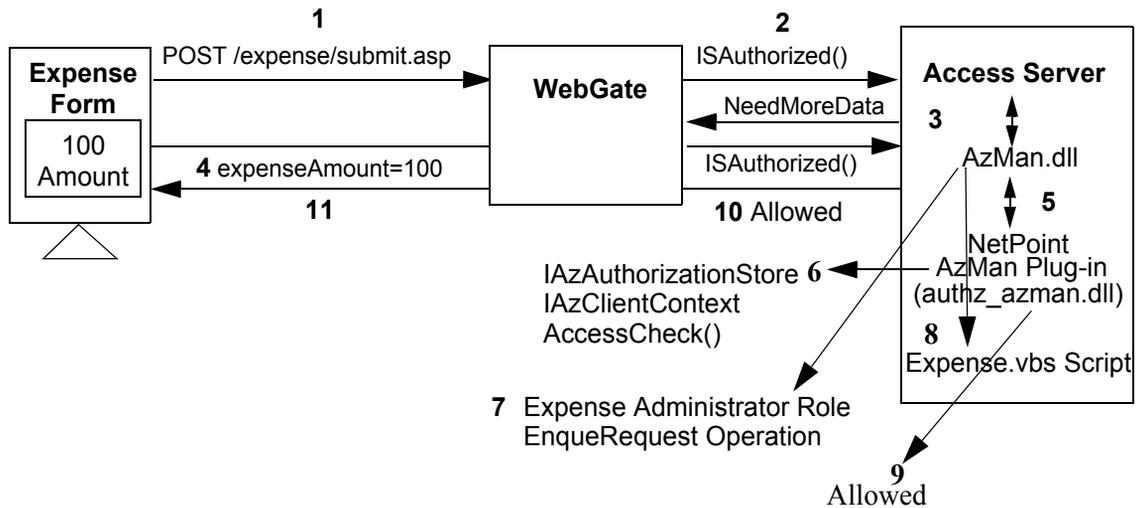
The authorization flow using the example that was implemented above is shown and described next.

## Example 3: NetPoint AzMan Plug-In Authorization Process Flow

The scenario below walks you through the authorization process, which is the same no matter where the Authorization Manager resides. In the following scenario:

- An Expense application was implemented to use Web forms to input expense data, as explained under “Example 1: An Expense Application” on page 521.
- An XML file is used for Authorization Manager policies, rather than storing these policies in the Active Directory. Both methods are valid, as described in “Authorization Stores” on page 516.
- The resource is protected by a NetPoint policy with an authorization rule based on a custom authorization scheme that passes parameters to the NetPoint

AzMan Plug-in, as described in “Using the NetPoint AzMan Plug-In with the Access Server API” on page 536.



### Process overview: NetPoint AzMan authorization after a user is authenticated

1. The user (user1k1) submits an expense form to /expense/submit.asp using the Authorization Manager Submit Expense task.
2. The WebGate intercepts the POST request to /expense/submit.asp and sends an ISAuthorized() request for user1k1, and the URL, to the Access Server.
3. The Access Server passes the parameter to the AzMan.dll, which applies the Submit Expense Policy, determines the expenseAmount variable is needed, and returns a NeedMoreData response to the WebGate.
4. The WebGate retrieves expenseAmount=100 from the POST data and re-sends the ISAuthorized() request with the data to the Access Server.
5. The Access Server:
  - a) Applies the Submit Expense Policy again
  - b) Executes the AzMan Plug-in for the Expense Authorization Scheme
  - c) Passes the expenseAmount variable in the RequestContext data and the samaccountname for the user in the Requestor data
6. The NetPoint AzMan Plug-in:
  - a) Constructs an IAzAuthorizationStore object, which in this case is for msxml://C:\MyAzStore.xml
  - b) Constructs an IAzApplication object for the Expense application
  - c) Constructs an IAzClientContext object for user1k1

- d) Calls AccessCheck() for the client context with the following:  
bstrobjectName = /expense/submit.asp  
varScopeNames = {}  
varOperations = {EnqueueRequest}  
varParameterNames = {expenseAmount}  
varParameterValues = {100}  
varInterfaceNames = {}  
varInterfaceFlags = {}  
varInterface = {}
7. The Authorization Manager runtime:
  - a) Determines that user1k1 is assigned to the Expense Administrator role  
The Expense Administrator role can perform the Submit Expense and Approve Expense tasks and includes the EnqueueRequest operation.
  - b) Determines that user1k1 is allowed to perform the EnqueueRequest operation
8. The Authorization Manager executes the expense.vbs script with expenseAmount=100, the script tests expenseAmount < 500 and returns a BusinessRuleResult of TRUE.
9. The NetPoint AzMan Plug-in receives the allowed result and returns Allowed.
10. The Access Server returns an allowed response to WebGate.
11. WebGate allows the POST request processing to proceed.

## Configuring the NetPoint AzMan Plug-In

The information below is provided to guide you during the configuration needed to use the NetPoint AzMan Plug-in. Some information is tailored for the Expense example discussed earlier. Your specifications may be different. Sample screens are presented in “Example 2: NetPoint Configuration” on page 525.

### **Task overview: Configuring the NetPoint AzMan Plug-in**

1. “Preparing Your Environment” on page 532
2. “Creating an Authorization Scheme for the AzMan Plug-In” on page 532
3. “Protecting Resources” on page 533
4. “Defining Authorization Rules and Policies” on page 534
5. “Using the NetPoint AzMan Plug-In with the Access Server API” on page 536

For a process overview, see “Using the NetPoint AzMan Plug-In with the Access Server API” on page 536

## Preparing Your Environment

The following procedures must be completed before you begin.

### Task overview: Preparing your environment

1. Install and set up Windows Server 2003 on the machine that will host the Access Server, as described in your Microsoft documentation.
2. Install and set up NetPoint, as described in the *NetPoint 7.0 Installation Guide*.

The NetPoint AzMan Plug-in is included with the Access Server, as discussed under “NetPoint Components and Requirements” on page 512.

3. Set up the AzMan authorization store, `azman.msc`, as described in your Microsoft documentation.

---

**Note:** By default, the group “Domain Admins” is listed within the Security tab when you create the Active Directory authorization store. The Access Server user (for example, Administrator) should also be listed in the Users and Groups list within the Security tab. Similar settings are *not* required for the XML store.

---

4. Design your application using the Authorization Manager to specify operation, task, and role definitions and role assignments, as described in your Microsoft documentation.

## Creating an Authorization Scheme for the AzMan Plug-In

The following steps presume that you have already defined an authentication scheme for this policy domain. The authorization scheme that you create can be included with any policy domain or policy and must include an authorization rule.

When you create a custom authorization scheme be sure to enter the full path to the shared `authz_azman` library (without the extension). You must also specify the user profile attribute values to be passed to the plug-in with the `RequesterInfo` data structure (the username is used to construct the `IAzClientContext` object). Also, specify the NetPoint AzMan Plug-in parameters needed for your own application. For more information about policy domains and authorization rules, see the *NetPoint 7.0 Administration Guide Volume 2*.

### To create a custom authorization scheme

1. Navigate to the Access System Console:

`http://hostname:port/access/oblix`

2. Navigate to the Authorization Management page: Access System Console > Access System Configuration > Authorization Management.
3. Click the Add button to begin a custom authorization scheme.
4. Enter the information for your custom authorization scheme.

For example:

**Name**—Name of this custom authorization scheme

**Description**—Optional description.

**Shared Library**—Full path to the authz\_azman library (without the extension) c:\NetPoint\access\oblix\lib\authz\_azman

**User Parameter**—samaccountname  
RA\_expenseAmount  
(The RA\_expenseAmount is a reverse action user parameter that is needed only if the authorization rule expects parameters).

**Optional Parameters**—

AzLogLevel	medium
AzApplication	Expense
AzRuleParameters	
AzOperations	
AzStore	msxml://C:\MyAzStore.xml

5. Save the scheme, as usual.

## Protecting Resources

You need to create a policy domain and add resources to protect. For general information about policy domains, see the *NetPoint 7.0 Administration Guide Volume 2*.

### To create a policy domain and add a resource

1. Navigate to the Access Manager, as usual:  
`http://hostname:port/access/oblix`
2. Create a policy domain, as usual: Access Manager > Create Policy Domain.

For example:

**Name**—Submit Expense

**Description**—Optional

---

**Note:** Do not enable the policy domain until you have finished all specifications for it, as described next.

---

3. Click Save.
4. Add a resource to protect with this policy domain: Access Manager > My Policy Domains > *link* > Resources > Add

For example:

**Resource Type**—http  
**URL Prefix**—/expense  
**Description**—Optional

5. Click Save.

## Defining Authorization Rules and Policies

You need to add the custom authorization scheme you created earlier to an authorization rule. The following steps presume that you have already defined your authentication rule for this policy domain.

### To add the authorization scheme to the authorization rule

1. Navigate to Authorization Rules page: Access Manager > My Policy Domains > *link* > Authorization Rules.
2. Click the Add button to display the Create Authorization Expression page.
3. Select Custom Authorization Scheme from the list, then click Add.

For example:

**Authorization Scheme**—Custom Authorization Scheme

A new page appears where you enter details for this rule.

4. Enter the details for this NetPoint authorization rule, and confirm that the authorization scheme you created earlier is selected in the Authorization Scheme list.

For example:

**Name**—Expense Authn  
**Description**—Optional  
**Authorization Scheme**—Expense Authorization

5. Save the rule, as usual.
6. Click Plug-in Parameters and confirm the profile attributes to be passed to the plug-in from the authorization scheme match those you specified in your custom authorization scheme.

For example:

**Profile Attributes Passed to Plug-In**— samaccountname

Optional Parameters—	Name	Value
	AzLogLevel	medium
	AzApplication	Expense
	AzRuleParameters	No Value
	AzOperations	No Value
	AzStore	msxml

7. Modify and save, if needed.
8. Add timing considerations and actions, as needed for your application.

### **To add default rules and the authentication rule**

1. Click the Default Rules link.
2. Click the Add button on the Default Rules page to add an authentication rule, which includes an authentication scheme.
3. Enter the details and save as usual.

For example:

**Name**—AzMan Basic Over LDAP

**Description**—Optional

**Authentication Scheme**—Basic Over LDAP

### **To add access policies**

1. Click the Policies link so you can add an access policy for the application
2. Click the Add button on the Policies page.

3. Fill in the requested information for your application and policy domain.

For example:

**Name**—SubExp Access Policy

**Description**—Refines control of the resource

**Resource type**—http

**Resource Operations**—GET  
POST

**Resource**—All

**URL Prefix**—/expense

**URL Pattern**—/expense/submit.asp

This policy contains no query string or query string variables.

4. Save the policy, as usual.

Delegating Administration is done as usual. There are no special requirements for the application in this example. For more information, see the *NetPoint 7.0 Administration Guide Volume 1*.

5. Click the General tab and enable the policy domain, as usual.

You request the resource as usual and NetPoint will complete the authorization process as described under “Example 3: NetPoint AzMan Plug-In Authorization Process Flow” on page 529.

## Using the NetPoint AzMan Plug-In with the Access Server API

The example below is provided as a guide if you want to use the Access Server API with the NetPoint AzMan Plug-in. For general information about the Access Server API, see the *NetPoint 7.0 Developer Guide*.

```
// Set up the Expense resource.
```

```
ObResourceRequest rr = new ObRequestRequest("http", "/  
expense/submit.asp", "POST");
```

```
// Authenticate DOMAIN\jsmith.
```

```
Hashtable creds = new Hashtable();  
creds.put("username", "user1k1");  
creds.put("password", "obl1x");  
ObUserSession user = new ObUserSession(rr, creds);
```

```
// Check if administrator is authorized to submit an expense  
form with expenseAmount=100.
```

```
// This uses the AzStore and AzApplication parameters defined  
by the Expense
```

```

// Authorization scheme and the AzOperations and
AzRuleParameter expenseAmount
// defined by the Submit Expense Authorization Rule.
//
// Equivalent access_test_cplus command:
// user1k1 oblix GET http://dotnet/expense/submit.asp dotnet
expenseAmount=100

Hashtable parameters = new Hashtable();
parameters.put("expenseAmount", "100");
if (user.isAuthenticated(rr, parameters)) {
    // authorized
}
else {
    // not authorized.
}

// Check if administrator is authorized to perform the
UseFormControl operation in
// the Expense application. This uses the AzStore and
AzApplication parameters
// defined by the Expense Authorization scheme but overrides
the AzOperations
// parameter in the Submit Expense Authorization Rule.
//
// Equivalent access_test_cplus command:
// user1k1 oblix GET http://dotnet/expense/submit.asp dotnet
//     expenseAmount=100&AzOperations=UseFormControl

parameters.put("AzOperations", "UseFormControl");
if (user.isAuthenticated(rr, parameters)) {
    // authorized
}
else {
    // not authorized.
}

// Check if the Expense Administrator role is authorized to
perform the
// UseFormControl operation in the Expense application. Note
that user1k1 must

```

```
// have this role.
//
// Equivalent access_test_cplus command:
// user1k1 oblix GET http://dotnet/expense/submit.asp dotnet
//
expenseAmount=100&AzRole=Expense+Administrator&AzOperations
=UseFormControl
//
// Note that access_test_cplus does not actually convert + to
blank, but it should.

parameters.put("AzRole", "Expense Administrator");
if (user.isAuthorized(rr, parameters)) {
    // authorized
}
else {
    // not authorized.
}
}
```

## Troubleshooting

An “Insufficient access right” error may appear in the log file when the Access Server user (for example, Administrator) does not appear in the Security tab of the Active Directory authorization store.

By default, the group “Domain Admins” is listed within the Security tab when you create the Active Directory authorization store. To run the Authorization Manager policy through NetPoint, the Access Server user (for example, Administrator) should also be listed in the Users and Groups list in within the Security tab. However, similar settings are not required for the XML store.

# 13 Integrating the NetPoint Security Connector for ASP.NET

NetPoint 7.0 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. This chapter explains 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. This chapter includes the following topics:

- “About ASP.NET” on page 539
- “About the Security Connector for ASP.NET” on page 542
- “NetPoint Components and Requirements” on page 542
- “Authorization with the Security Connector for ASP.NET” on page 543
- “Using the Security Connector for ASP.NET” on page 544
- “NetPoint Role-Based Authorization” on page 549

## About ASP.NET

ASP.NET is a set of technologies in the Microsoft .NET Framework that enables the building of Web applications and XML Web services using compilation and caching technologies available in the .NET Framework. ASP.NET pages:

- Execute on the server and generate markup such as HTML, WML, or XML that is sent to a browser
- Use a compiled, event-driven programming model that enables the separation of application logic and the user interface
- Contain server-side logic, rather than client-side logic, written in Visual Basic .NET, C# .NET, or any .NET-compatible language

Developers can use the .NET Framework class library, which is an object-oriented collection of reusable types, to create ASP.NET applications. Web applications and XML Web services benefit from features of the common language runtime (CLR).

## Security Principals and Security Identifiers (SIDs)

Both ASP.NET and Microsoft Internet Information Services (IIS) provide security models that allow you to authenticate users appropriately and obtain the correct security context within your application.

The user's (or potentially an application's or computer's) identity is referred to as a security principal. The client must provide credentials to allow the server to verify the identity of the principal. After the identity is known, the application can authorize the principal to access protected resources.

Windows provides a `WindowsPrincipal` object that defines a user identity and the user's role identity. The role identity is the role or roles defined in Windows for the user identity. Microsoft .NET technology provides an interface to create a `Principal` object using only Windows-specific roles. ASP.NET applications can call the `WindowsPrincipal.IsInRole` method to find out if the identity is in a specific role, for example, the admin role or users role.

Security within the ASP.NET framework revolves around security identifiers (SIDs). SIDs are equal to NetPoint SSO tokens and represent a unique user within the Windows operating system. ASP.NET wraps each SID into a series of managed objects that allow a developer to impersonate that user.

The main object that wraps the SID is the `Identity` object (`Identity`). This object allows a developer to discover how that identity was established by calling methods to obtain:

- The authentication method
- The name of the identity
- The authentication status (authenticated or not)

For more information, consult the Microsoft ASP.NET documentation.

## `IPrincipal.IsInRole` Method Syntax

A principal object represents the security context of the user on whose behalf the code is running, including that user's identity (`Identity`) and any roles to which the user belongs. The .NET framework class library `IPrincipal` interface defines the basic functionality of a principal object.

---

**Note:** All principal objects are required to implement the `IPrincipal` interface.

---

During the authorization process, the public `IPrincipal.IsInRole` method determines whether the current principal belongs to the specified role.

The following `IPrincipal.IsInRole` method syntax is based on .NET Framework version 1.1 and intended only as an example:

```
[Visual Basic]
Function IsInRole( _
    ByVal role As String _
) As Boolean
[C#]
bool IsInRole(
    string role
);
[C++]
bool IsInRole(
    string* role
);
[JScript]
function IsInRole(
    role : String
) : Boolean;
```

## Parameters

*role*

The name of the role for which to check membership.

## Return Value

**true**—Returns true if the current principal is a member of the specified role.

**false**—Returns false otherwise.

## Requirements

This operates on all Windows platforms that support the .NET framework. NetPoint 6.5 WebGates operate on the Windows 2000 Server and Windows Server 2003 family.

---

**Note:** Older NetPoint WebGates are compatible with NetPoint 7.0 Access Servers. However, older NetPoint WebGates use a different encryption scheme for the shared secret as discussed in the *NetPoint 7.0 Administration Guide Volume 2*.

---

For more information and the most current syntax, see the Web site:

<http://msdn.microsoft.com/developercenters/>

## About the Security Connector for ASP.NET

The NetPoint Security Connector for ASP.NET:

- Provides a dynamically-loaded native library assembly to enhance ASP.NET behavior to take advantage of NetPoint features
- Extends pre-defined Microsoft roles to include dynamic groups and any attribute values defined in a NetPoint user's profile

For example, Microsoft provides pre-defined roles within a Windows domain; however, Microsoft roles do not include the flexibility of NetPoint's dynamic groups, timing, and other conditions that can alter a user's access rights.

You can customize your ASP.NET application or Web service to use the NetPoint assembly during the authorization process. This converts NetPoint authorization actions into roles using a header variable that maps to roles that are meaningful to the .NET environment.

---

**Note:** Administrators must plan and coordinate the roles that will be used with the application developer or deployer. NetPoint does not “know” what the .NET roles are, and has no way to discover what the roles are. The NetPoint role and the .NET role are related only via a NetPoint role string. There is no referential integrity supplied or implied.

---

## NetPoint Components and Requirements

The NetPoint Security Connector for ASP.NET library assembly, `OBPrincipalHTTPModule.dll`, is installed with the WebGate in the same directory as the `webgate.dll`. For example:

```
\WebGate_install_dir\access\oblix\apps\webgate\bin\  
OBPrincipalHTTPModule.dll
```

Because more than one application may share the `OBPrincipalHTTPModule` assembly, it is included in the global assembly cache (GAC).

The new NetPoint library assembly runs as an ASP.NET `HttpModule`. Therefore, you must include details about this assembly in the `Web.config` file on the same machine and in the same directory as the ASP.NET application.

The `OBPrincipalHTTPModule` assembly includes a new `OblixPrincipal` object class and the `OblixHttpModule`. With this assembly, the ASP.NET application can define and pass a `NetPoint` role. Without this assembly, the application can pass only Windows roles.

## The OblixHttpModule

The `OblixHttpModule` recognizes `NetPoint` roles, and the roles supported with the `WindowsPrincipal` object. In fact, `NetPoint` recognizes any principal object in the .NET framework class library.

The `OblixHttpModule` must be specified as an action type in the `NetPoint` authorization rule that protects the ASP.NET application. For details, see “Setting Up the `NetPoint` Role Action” on page 548.

During authorization, the `OblixHttpModule`:

- Instantiates the new `OblixPrincipal` object before control is given to the ASP.NET application using the `WindowsPrincipal` object (or any other type of principal object in the .NET framework class library) that was extracted from the request.
- Adds `NetPoint` Role data (the `NetPoint` authorization rule action) to the `OblixPrincipal` object.
- Associates the `OblixPrincipal` object with the ASP.NET HTTP request.

---

**Note:** Whenever the application requests the principal object, it receives the `OblixPrincipal` object, which encapsulates all other principal objects.

---

## The OblixPrincipal Object

A role is the name of a membership category, for example `admin` or `user`. The `OblixPrincipal` object represents the security context of the user on whose behalf the code is running. This includes the user's identity (`IIdentity`) and any roles to which they belong as derived from the `WindowsIPrincipal` interface. The `iPrincipal.IsInRole` method checks both `NetPoint` roles and the `iPrincipal` interface.

## Authorization with the Security Connector for ASP.NET

The following high-level overview introduces authorization using the `NetPoint` Security Connector for ASP.NET. For a more detailed example, see “`NetPoint` Role-Based Authorization” on page 549.

## **Process overview: Authorization with the NetPoint Security Connector for ASP.NET**

1. After the user is authenticated, the WebGate begins the authorization process with the Access Server, as usual.
2. The IIS Web server creates the WindowsPrincipal object based on Windows Impersonation.
3. The OblixHttpModule receives the request, instantiates a new OblixPrincipal object using the WindowsPrincipal object, and adds NetPoint Role data based on the NetPoint authorization rule action.
4. The OblixPrincipal object initializes itself, then recognizes and stores NetPoint role data in memory.
5. The OblixHttpModule associates the OblixPrincipal object with the request and returns control to the IIS Web server.
6. The ASP application extracts the OblixPrincipal object for the request and calls the IPrincipal.IsInRole method.
7. The OblixPrincipal object calls the WindowsPrincipal object's IsInRole method from the .NET Framework Class Library to determine whether the current principal belongs to the specified .NET role, checks the NetPoint role, and returns the answer to the ASP.NET application.
8. The IPrincipal.IsInRole method returns true or false, depending on the current user's identity and the NetPoint authorization rule. For a syntax example, see "IPrincipal.IsInRole Method Syntax" on page 540.

If the answer is false, the Principal object looks in the NetPoint role list for the requested role and returns the answer to the ASP.NET application. If the answer is true, the ASP.NET application completes processing and access to the resource is granted.

## **Using the Security Connector for ASP.NET**

The following task overview explain how to use the NetPoint Security Connector for ASP.NET.

### **Task overview: Setting up the NetPoint Security Connector for ASP.NET**

1. Install a NetPoint WebGate, as described in "Setting Up Your Environment" on page 545.
2. Setup the application, as described in "Setting Up the ASP.NET Application for the NetPoint Security Connector" on page 546.

3. Configure the NetPoint role action, as described in “Setting Up the NetPoint Role Action” on page 548.

For a process overview, see the “NetPoint Role-Based Authorization” on page 549

## Setting Up Your Environment

Before you can use the NetPoint Security Connector for ASP.NET, you must set up the NetPoint WebGate on a machine hosting the IIS Web server and the .NET framework with ASP.NET.

You are given the option to include the .NET framework and ASP.NET during IIS Web server installation. This automatically configures the IIS metabase. The ASP.NET application security configuration and the IIS security configuration are independent. Each may be used separately or together.

IIS maintains security related configuration settings in the IIS metabase. ASP.NET maintains security (and other) configuration settings in XML configuration files. For more information, see your Microsoft documentation.

### To set up your environment

1. Install the IIS Web server and the .NET framework v1.1 with ASP.NET.

---

**Note:** If you are using IIS v6.x, be sure to enable ASP.NET applications. The actions in step 2 occur automatically if you install the .NET framework after installing the IIS Web server. If this reflects your installation, skip to step 3.

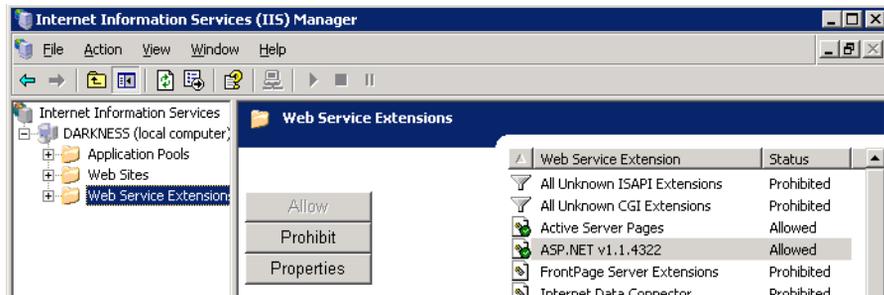
---

2. Register ASP.NET and allow the ASP.NET Web Service Extension on the machine that will host the NetPoint WebGate, if needed.

For example:

```
aspnet.regiis.exe
```

```
IIS > local_host > Web Service Extensions > ASP.NET v1.1.4322 > Properties  
> Allowed
```



3. Install the NetPoint 7.0 WebGate on a machine hosting the IIS Web server, the .NET framework, and ASP.NET.

The NetPoint WebGate installation will terminate before completion if the .NET framework is not included on the WebGate host. To share the OBPrincipalHTTPModule assembly among all applications, the assembly is installed as part of the global assembly cache.

## Setting Up the ASP.NET Application for the NetPoint Security Connector

When you create the ASP.NET application or Web service using Visual Studio .NET, a generic Web.config file is created automatically. You can modify this file to customize your application to use the OBPrincipalHTTPModule.dll assembly during authorization.

With impersonation enabled, ASP.NET applications can execute with the identity of the client on whose behalf they are operating. ASP.NET will receive the token to impersonate from IIS.

---

**Note:** If you do not enable impersonation in the application and in NetPoint, Oblix roles will be returned, which encapsulate all roles. However, the permissions may not be all you need.

---

### To set up the ASP.NET application

1. Use Visual Studio .NET to write your ASP.NET application or Web service, as described in the Microsoft documentation.
2. Include the details below in the Web.config file under <System.Web> to use the OblixHttpModule and OBPrincipalHTTPModule.dll.

Be sure to include your own PublicKeyToken. For example:

```
-->
<httpModules>
  <add type="Oblix.Agents.OblixHttpModule,
ObPrincipalHttpModule, Version=7.0.0.0, Culture=neutral,
PublicKeyToken=xxxxxxxxxxxx" name="OblixHttpModule"/>
</httpModules>
<compilation
  defaultLanguage="c#"
  debug="true"
/>
```

---

**Note:** The value of Culture= is case sensitive; “neutral” must be lowercase.

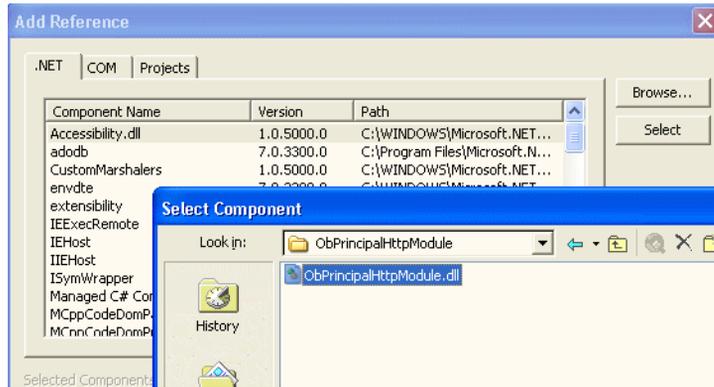
---

3. Reference the ObPrincipalHttpModule assembly in your application: right-click the project in Visual Studio, select Add Reference, click the Project tab, then browse for and select the global assembly cache.

---

**Note:** You may also use the /r option if the application is built from the command line.

---



You can either reference the IPrincipal object in the application, as described below, or point to the ACLs in the Web.config file.

4. Add the IPrincipal references for the required .NET assemblies to the application (or see the sample web.config file).

For example:

```
using System.Security.Principal;  
using System.Web.Security;
```

5. Add a method to the application that calls the IPrincipal.IsInRole() function with the appropriate parameters for your application.

For example, if the return value of the authorization-rule action is Manager, the method would be:

```
Context.User.IsInRole("Manager"); // Context - HttpContext  
object associated with the page  
// user - Principal object
```

The application is now set up to use the NetPoint Security Connector. Next you must set up the NetPoint Role action.

## Setting Up the NetPoint Role Action

Actions can pass information about users to other applications in the same, or different, NetPoint policy domain. Authorization actions occur when a user requests access to a resource (that is, when the user requests the resource's URL).

Before passing roles to the `OblixHttpModule`, you need to create a Role action in the authorization rule for the policy domain that protects the ASP.NET application. This action relies on the `OblixHttpModule`. Aware NetPoint clients can use the Role action to define roles separate from header variables.

The Role is contained in a Principal object. You may have as many roles as you choose. Each Role value will be added to the `OblixPrincipal` object. When calling the `IPrincipal.IsInRole` method from the .NET Framework Class Library with any of the Role values, true is returned.

### To set up NetPoint Role Actions

1. Create a policy domain in the NetPoint Access Manager to protect the ASP.NET applications and include an impersonation action.

For more information on policy domain configuration and single sign-on configuration, see the *NetPoint 7.0 Administration Guide Volume 2*.

2. Select the authorization rule and click the Actions tab to define an action for this authorization rule to pass roles to the `OblixHttpModule`:

- **Type**—Specify `OblixHttpModule`. Only a type of `OblixHttpModule` will be forwarded to the ASP.NET `HttpModule`.
- **Name**—Supply a role or a name.

You can have as many roles as you choose. Each role value is added to the `OblixPrincipal` object. When calling the `IPrincipal.IsInRole` method with any of these values, true is returned.

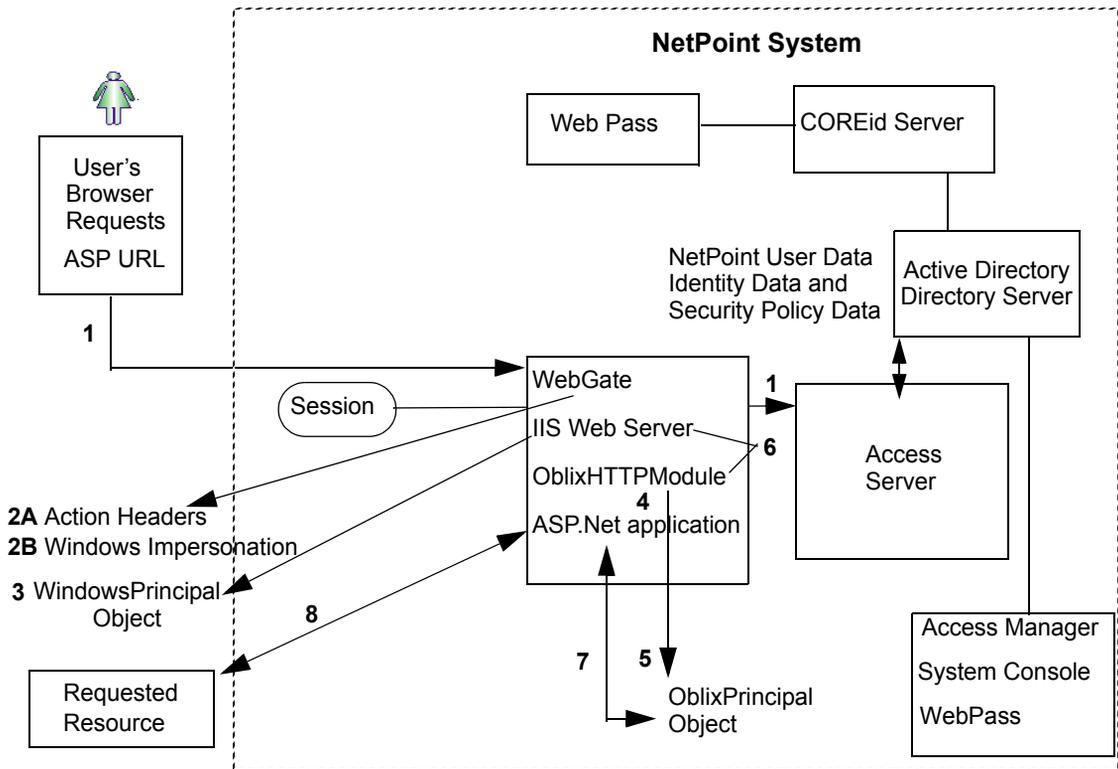
Only one name may be set. Name exists in the event that IIS does not provide an Identity name, for example, None authentication in the `Web.config` file. If Windows authentication is set and the `WebGate` is configured for impersonation, this action is ignored.

- **Return Value**—This can be any static or dynamic value, like any other action.
3. Save the rule and restart the Access Server to have your changes take affect. Your environment is set up, the ASP.NET application is complete, and the NetPoint policy domain protects the application with the new authorization rule.

# NetPoint Role-Based Authorization

The following process occurs during authentication and role-based authorization with the NetPoint Security Connector for ASP.NET. Figure 15 illustrates the sequence and is followed by a detailed description.

**Figure 15** NetPoint Security Connector for ASP.NET Authorization Flow



## Process overview: Events during authentication and authorization

1. The Web server receives the user's ASP URL request. The WebGate intercepts the request and communicates with the Access Server to determine:
  - If the resource is protected
  - How the resource is protected
  - If the user is authenticated
  - If user access is authorized

Authentication is performed between the Access Server and directory server, as usual.

2. When the user is authenticated, the WebGate begins the authorization process with the Access Server and:
  - a) Sets action headers for roles
  - b) Performs Windows Impersonation
  - c) Returns control to the IIS Web server
3. The IIS Web server creates the WindowsPrincipal object based on Windows Impersonation.
4. The OblixHttpModule:
  - a) Receives the request
  - b) Instantiates a new OblixPrincipal object using the WindowsPrincipal object that was extracted from the request
  - c) Adds NetPoint Role data (the authorization rule action)
5. The OblixPrincipal object initializes itself and stores NetPoint role data in memory.
6. The OblixHttpModule associates the OblixPrincipal object with the request and returns control to the IIS Web server.
7. The ASP application extracts the OblixPrincipal object for the request and calls the IPrincipal.IsInRole method. The OblixPrincipal object:
  - a) Calls the WindowsPrincipal object's IsInRole method
  - b) Checks the NetPoint role
  - c) Returns the answer to the ASP.NET application

If the answer is false, the Principal object looks in the NetPoint role list for the requested role and returns the answer to the ASP.NET application.
8. The ASP.NET application completes processing and access to the resource is granted.

# 14 Integrating Workflows With MIIS Provisioning

This chapter describes how to use the NetPoint 7.0 COREid System with the Microsoft Identity Integration Server (MIIS) provisioning product.

This chapter covers the following topics:

- “About Provisioning with MIIS” on page 552
- “Integration Architecture” on page 553
- “About Integrating NetPoint and MIIS” on page 555
- “Preparing Your Environment” on page 556
- “Installing the Oblix Management Agent” on page 558
- “Configuring the Oblix Management Agent” on page 562
- “Copying the MIIS Files to COREid” on page 570
- “Configuring the Database, COREid, and SSO” on page 571
- “Configuring Logging” on page 580
- “About the ObMA Application” on page 581
- “Using a COREid Workflow to Delete Identities from Metaverse” on page 582

# About Provisioning with MIIS

MIIS enables organizations to manage issues that arise from storing identity information in different data sources. These issues include the administrative overhead associated with duplicated information and the need to synchronize information from different data sources. MIIS manages how data is passed among different data sources. MIIS selectively imports identity information from each data source, determines which directory contains authoritative data for specific purposes, and updates other data sources based on the data determined to be authoritative.

With NetPoint 7.0, you can use COREid System workflow functionality to add, modify, and delete information about users, and to propagate this information to different target data sources that are provisioned using MIIS. For example, you can configure an Add or Modify User workflow in COREid to provision user accounts in Exchange.

You can use the COREid System as the data entry mechanism for provisioning user accounts in applications controlled by MIIS. For example, if you add or delete a user via a COREid workflow, that user's identity can also be added or deleted from any of the data sources managed by your MIIS system. In this way, a COREid application such as the User Manager can act as a central point of administration for provisioning user application accounts.

The Oblix solution for provisioning using MIIS includes an object schema template and workflows that can be configured in the COREid System, along with agents and connectors that pass data from COREid to MIIS. The solution also includes a SQL Server Management Agent that you configure in MIIS.

Key terms for working with the Oblix Management Agent for MIIS include:

**Data Store**—A data repository.

**Connected Data Source**—A persistent repository for identity data. The connected data source belongs to another application, and it communicates with MIIS for the purpose of synchronizing data with other connected data sources.

**Metaverse**—A meta-directory that aggregates identity data from multiple Management Agents. The Metaverse provides a single, global, integrated view of all combined objects.

**Connector Space**—A staging area for synchronization of data to the Metaverse and export of data to the connected data source.

**Management Agent**—An agent that is responsible for importing and exporting data to and from MIIS. Each type of connected data source has a specific Management Agent (MA), for instance, there may be an MA for SQL Server and one for Active Directory. The MA translates a required operation into a format that the connected data source understands. Oblix uses a SQL Server MA to synchronize identities provisioned by COREid to the Metaverse.

**Schema Domain**—The scope of object and attribute schema, usually tied to a single directory or a set of directory servers of the same type.

**Synchronization**—The process by which data is moved from one connected data source to a target connected data source. Synchronization consists of reading data from the connected data source into the connector space (import and staging), propagating the data to the Metaverse (aggregation), and then propagating the data to the target (provisioning, de-provisioning, or exporting an attribute flow).

## Integration Architecture

The integration architecture enables two main functions:

- **Password Synchronization**—When a password is updated in COREid, password synchronization ensures that all appropriate connected data sources are updated in a secure manner.
- **Provisioning Data Other Than Passwords**—When any data other than a password is updated in COREid, this data is first aggregated in the Metaverse, then provisioned to the appropriate connected data sources.

When you install the NetPoint 7.0 COREid System, an MIIS connector is included in the installation. This connector is the intermediary between COREid and the Oblix Management Agent for MIIS (the Oblix MA).

Figure 16 illustrates one possible installation environment for this solution.

**Figure 16** Installation environment for the Oblix MA

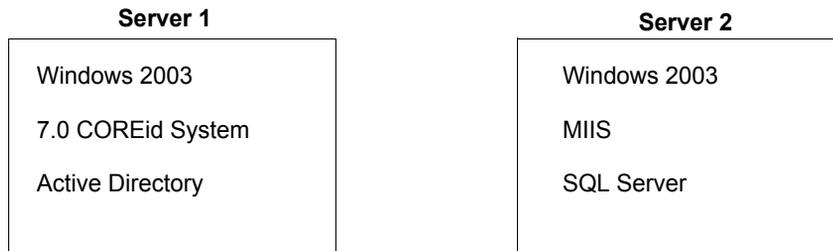
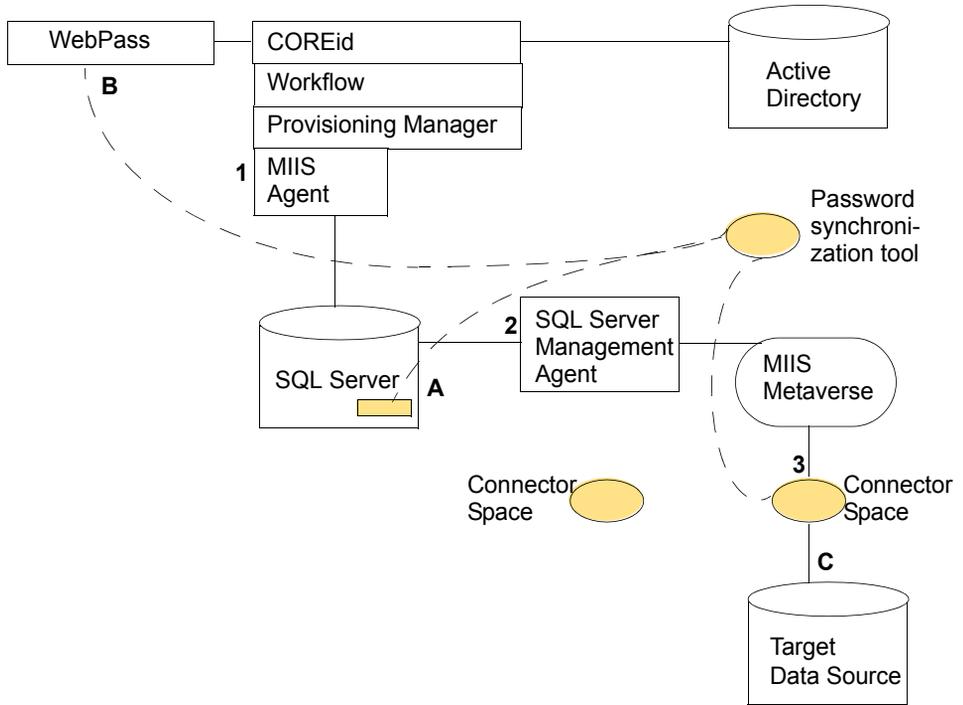


Figure 17 illustrates how data is sent from COREid to MIIS.

**Figure 17** Data architecture for provisioning to MIIS via COREid



### Process overview: NetPoint Provisioning with MIIS

1. The COREid System loads a template file that contains the MIIS Metaverse schema.  
  
When a user adds or updates identity data via a COREid workflow that uses this schema, the Oblix Provisioning Manager passes the data to an Oblix MIIS Agent.
2. The COREid MIIS Agent writes the identity data to a SQL server.
3. The identity data is synchronized to the target data repository through the Oblix SQL Server Management Agent.

## Process overview: Password synchronization

1. To ensure that passwords are transmitted securely, when a password is updated using a COREid workflow, the SQL server stores a workflow step instance ID in a table (see A in Figure 17).
2. The password synchronization tool (the ObMA application) retrieves the workflow step instance ID from the SQL Server, generates an IdentityXML request using HTTPS, and sends the request to the COREid Server via the WebPass (B in Figure 17).

When you install the Oblix MA, the ObMA application is created to handle password synchronization and cleanup.

The WebPass may be protected by a WebGate.

3. The OblixMA determines the target connectors on which to set the password (C in Figure 17).

If the MA for that connector has the Set Password function enabled, MIIS immediately updates the password in the connector.

---

**Note:** You must manually configure the secure connections used for passing password data. During installation, you set a flag that indicates the use of SSL for password synchronization, but you must then explicitly set up SSL between the machine hosting the ObMA application and the machine hosting the WebPass Web server.

---

## About Integrating NetPoint and MIIS

The following overview describes the high-level steps for setting up the integration between NetPoint and MIIS as well as where to find more information about each procedure.

### Task overview: Integrating NetPoint with MIIS provisioning

1. Complete all steps in “Preparing Your Environment” on page 556.
1. Install the Oblix Management Agent for MIIS, as described in “Installing the Oblix Management Agent” on page 558.
2. Configure the Agent in MIIS, as described in “Configuring the Oblix Management Agent” on page 562.
3. Copy the MIIS object template file and the metadata file to COREid installation directories, as described in “Copying the MIIS Files to COREid” on page 570.

4. Configure an RDBMS profile for the SQL Server that contains the OblixMA database, as described in “Configuring the Database, COREid, and SSO” on page 571.
5. Configure a license, as described in “Configuring the Database, COREid, and SSO” on page 571.
6. Configure the MIIS template objects and attributes in the COREid System Console, and associate the MIIS attributes with a COREid application tab, as described in “Configuring a COREid Workflow for MIIS Provisioning” on page 573.

---

**Important:** For MIIS password synchronization to work, you must configure an attribute with a semantic type of Password in the COREid System Console. The MIIS template (miis.tpl) contains a binary attribute, obattrbin, that may be configured as Password semantic type. For more information, see the discussion of this file under “Confirming Installation” on page 559.

---

7. Create a workflow that uses the MIIS object template attributes, as described in “Configuring a COREid Workflow for MIIS Provisioning” on page 573.

---

**Note:** Ensure that the workflow has two commit steps: one step that commits the object template data to the MIIS schema domain, and one that commits the data to LDAP.

---

8. If the ObMA application communicates with a WebPass that is protected by a WebGate, enable SSO between COREid and the SQL Server Management Agent and the console application as described in “Configuring the Database, COREid, and SSO” on page 571.

## Preparing Your Environment

The following prerequisites must be met before installing the Oblix Management Agent for MIIS.

### **Task overview: Preparing for Oblix Management Agent installation**

1. “Preparing for Oblix Management Agent Installation” on page 557.
2. “Preparing for Password Synchronization” on page 557

## Preparing for Oblix Management Agent Installation

An instance of Microsoft SQL Server 2000 is required as the connected data source for the Oblix MA. This does not have to be the same SQL Server instance that MIIS uses.

### To prepare for Oblix Management Agent for MIIS installation

1. Install Microsoft SQL Server 2000 (with SP 3, version 8.00.760).
2. Ensure that MIIS 2003 is running properly on Microsoft Windows 2003 Server.
3. Install, set up, and configure the NetPoint 7.0 COREid Server and WebPass.

---

**Note:** If you uninstall MIIS after installing the Oblix Management Agent, the OblixMA database can not be removed until you restart the SQL server. In this case, be sure to restart the SQL server and remove the OblixMA database manually.

---

## Preparing for Password Synchronization

The ObMA application uses the WMI Provider for MIIS to set passwords on target connector space objects. Only members of the MIISPasswordSet and the MIISAdmin security groups can perform this function. This means that the “RunAs” user for the ObMA application must be a member of these two groups. In addition, the ObMA application executes SQL statements against the OblixMA database. The “RunAs” user must also have permission to execute queries and updates to the OblixMA database.

MIIS provides a WMI Provider for password synchronization. The ObMA application uses the WMI Provider for MIIS for password synchronization.

The user for the scheduled tasks must be part of the “MIISAdmins”, “MIISPasswordSet”, and “MIISBrowse” security groups. The selected user must have permissions to execute the SQL queries and update statements to the OblixMA database and the MIIS database.

### To prepare for password synchronization

1. Ensure that the that the “RunAs” user for the ObMA application is a member of the MIISPasswordSet security group.
2. Ensure that the “RunAs” user for the ObMA application is a member of the MIISAdmins security group.
3. Ensure the “RunAs” user for the ObMA application has permissions to execute the SQL queries and update statements to the OblixMA database and the MIIS database.

# Installing the Oblix Management Agent

The installation package for the Oblix Management Agent for MIIS is:

NetPoint7\_0\_Win32\_Connector\_MIIS.exe

## To install the Oblix Management Agent for MIIS

1. Log in to the MIIS host machine using the identity of an administrator with SQL Server administrator privileges.
2. Run the installation package NetPoint7\_0\_Win32\_Connector\_MIIS.exe on the MIIS host machine.

Installshield starts up. After a few minutes, the Welcome screen appears.

3. Click Next to dismiss the Welcome screen.

The License Agreement screen appears.

4. Accept the license agreement, then click Next.

A warning screen appears.

5. Ensure that you are logged in with either Windows administrator or SQL server administrator privileges, then click Next.

The Select Installation Folder screen appears.

6. Select a folder where you want to install the Management Agent.

For example, the default is:

C:\Program Files\Oblix Management Agent

7. Click Next.

8. Configure the following items when asked:

**Server**—This is the name or fully qualified host name of the SQL server instance that contains the OblixMA database. This database is the connected data source for the Oblix MA.

**Windows or SQL authentication**—For SQL authentication, you must also enter the username and password that you use for authentication.

9. Click Next.

10. Supply the following information:

**Host**—The fully-qualified host name of the machine where the Web server that hosts the WebPass is installed.

---

**Important:** For SSO to work between the ObMA application and COREid, the host name must be fully qualified.

---

**Port**—The port number of the Web server instance that hosts WebPass.

**Enable SSL**—This option sets an SSL flag in the obmaconfig.xml file. See “Customize the Management Agent Configuration File” on page 577 for details. When this flag is set, ObMA sends IdentityXML over HTTPS. For this to work, SSL must be set up between the machine hosting the ObMA application and the WebPass Web server.

**Login name**—The administrator user ID that you will use when logging in to COREid.

**Password**—The administrator password that you will use when logging in to COREid.

**11.** Click Next.

The installer configures and writes the installation files.

**12.** Click Done.

**13.** Go to the SQL Server Enterprise Manager to verify that the OblixMA has been installed.

## Confirming Installation

You can confirm that all elements were installed by reviewing this discussion and comparing your installation with the details here.

The OblixMA consists of a SQL Server database, the ObMA application, supporting files for logging, Metaverse and Management Agent extensions, plus supporting assemblies. By default (unless you specified a different path during installation), the OblixMA files are located in the directory below:

C:\Program Files\Oblix Management Agent

Two scheduled tasks are included and can be viewed in the Windows Control Panel:

- Every 15 minutes, delta synchronization is executed, followed by password synchronization, then clean up of the delta table. The command line looks like this:  
obma -r “Delta Synchronization” -p -d oblix\_delta
- Once each day, clean up of the main table is executed. The command-line is:  
obma -d oblix\_main

The installed scheduled tasks are provided as examples of how the OblixMA Management Agent and the ObMA application may be used together to synchronize data, then passwords. It is expected that the frequency, RunAs user, and possibly the command-line may be changed to better suit the needs of a particular deployment.

---

**Important:** The user for the scheduled tasks must be part of the “MIISAdmins” and “MIISPasswordSet” security groups. The user selected must have permissions to execute the SQL queries and update statements to the OblixMA database and the MIIS database.

---

The following files are installed in the to COREid folder. Later on, you will copy these files to your COREid System installation directory, as explained in “Copying the MIIS Files to COREid” on page 570:

- **An Object Template File**—This file is named `miis.tpl`.

A template is a file-based schema definition. The COREid System loads templates upon start-up. You configure template-based objects and attributes in the COREid Administration Console. Once configured, you can associate these attributes with a tab and a workflow definition. Template-based attributes are accessible through Identity XML and the Identity Event API.

The `miis.tpl` file contains the MIIS Metaverse schema. As described in “Copying the MIIS Files to COREid” on page 570, you copy this `.tpl` file to `COREid_install_dir\oblix\config\template\miis.tpl`. You can then configure COREid applications and workflows using the objects and attributes defined in this file. See the *NetPoint 7.0 Administration Guide, Volume 1* for details.

The MIIS template (`miis.tpl`) contains a binary attribute, `obattrbin`, that may be configured as Password semantic type. `miis.tpl` is built from the Metaverse schema during installation of the COREid Connector for MIIS. Since passwords are not synchronized through the Metaverse there is no Metaverse attribute used to store passwords and therefore no appropriate attribute in `miis.tpl` with which to configure the password attribute in COREid. However, `obattrbin` can be used for this purpose.

- **A Metadata File**—This file is named `obmetadata.xml`. It provides table definitions for the objects and attributes defined by Oblix and those from the MIIS Metaverse. After configuring the OblixMA in MIIS, you copy this file to a COREid installation directory (`COREid_install_dir\oblix\config\obmetadata.xml`).
- **A License File**—This file is named `LICENSE.MIIS.tmp`. It provides a temporary license that allows you to use the OblixMA for 30 days. You obtain a permanent license key from Oblix, and copy the license string in the COREid System Console. See “Copying the MIIS Files to COREid” on page 570 for details.

The following files are created in the logs folder the first time that the corresponding application is run:

- **obma.log**—The ObMA application writes logs to this file.
- **oblog.log**—The installer uses this file to capture installation information. For information on the format and configuration of the oblog.log file, see the chapter on logging, auditing, and SNMP in the *NetPoint 7.0 Administration Guide, Volume 1*.
- **obmaextension.log**—The ObMASynchronization and ObMVSynchronization extensions log to this file.

The following files are installed in the OblixMA extensions folder. The installer also copies these files to the Microsoft Identity Integration Server extensions folder:

- **Extensions**—These files are used when you create programs to write information from COREid to MIIS.
  - **obmasynchronization**—This extension implements the IMASynchronization interface for the OblixMA Management Agent.
  - **obmvsynchronization**—This extension implements the IMVSynchronization interface and contains the necessary functionality in order for deletes to be provisioned from the OblixMA Management Agent. The MIIS implementation of IMVSynchronization must invoke the corresponding methods of IMVSynchronization from obmvsynchronization.
  - **Supporting Assemblies**—For obmasynchronization and obmvsynchronization the supporting assemblies include:
    - obmalib.dll
    - obmautil.dll
    - obmasynchronizationimpl.dll
    - obmvsynchronizationimpl.dll

The following file in the config folder is of interest:

- **obmaconfig.xml**—This file is used by the password synchronization tool and by the MA and Metaverse extensions for the MIIS server.  
See “Customize the Management Agent Configuration File” on page 577 for details.

# Configuring the Oblix Management Agent

After installing the Oblix Management Agent (MA) for MIIS, you need to configure the MA in MIIS. The OblixMA is configured against the SQL Server 2000 instance where the OblixMA database (named OblixMA) is located. The following procedures describe the wizard pages that you step through while configuring the MA.

Before configuring the SQL Server Management Agent, you should be familiar with the use of join rules, projection rules, and attribute flows in MIIS. Join and Projection rules define how to link identities from COREid to the Metaverse. Attribute flows determine which attributes are placed in the Metaverse and then synchronized to target connector spaces.

You should also understand run profiles in MIIS. Run profiles define the process that you want to use to synchronize data from the connected data source to the target connector spaces.

You should also be familiar with attribute precedence as defined in the MIIS Metaverse Designer. Attribute precedence determines which value supersedes other values when multiple Management Agents have attribute flows to the same Metaverse attribute.

Finally, you should have a plan for what attribute values you wish to send from a COREid workflow to MIIS. This will determine what join and projection rules you create, and how you wish to configure attribute flows in MIIS.

The following provides an overview of the high-level steps to configure a management agent and where to find more information.

## **Task overview: Configuring the Management Agent**

1. Create a Management Agent, as described in “Creating a Management Agent in MIIS” on page 563.
2. Establish a connection to a connected data source, as described in “Configuring the SQL Server Connection” on page 563.
3. Choose elements that will participate in synchronization, including object classes, attributes, and database table columns, as described in “Configuring the MIIS Database Columns” on page 564.
4. Define synchronization rules, as described in “Configuring Join and Projection Rules” on page 565.
5. Target attributes for each identity that you are mapping, as described in “Configuring the Attribute Flow” on page 567
6. Set up deprovisioning, as described in “Configuring Deprovisioning” on page 569.

7. Point to an MA extension, as described in “Configuring Extensions” on page 569.
8. Enable password synchronization, as described in “Password Synchronization” on page 578.

---

**Note:** Refer to the MIIS online help for background on key topics such as join and projection rules, attributes flows, and run profiles. You will need to understand these concepts before configuring the Oblix MA. The MIIS online help provides in-depth coverage of these topics.

---

## Creating a Management Agent in MIIS

The first procedure you complete is creating a management agent.

### To create the Management Agent in MIIS

1. From the Windows Start menu, open the MIIS Identity Manager.
2. Click the Management Agents button in the toolbar.
3. In the Actions list, click Create.

The Create Management Agent wizard starts up.

4. In the first page of the wizard, in the “Management agent for” field, select the following option:

SQL Server 7.0 or 2000

5. In the Name field, enter OblixMA.
6. In the Description field, enter a description.
7. Click Next.

The Connect to Database page is displayed.

8. Continue with “Configuring the SQL Server Connection” on page 563.

## Configuring the SQL Server Connection

When configuring the SQL server connection, you will enter the user name and password that you use for the SQL server or for Windows authentication, as appropriate:

- **Windows Authentication**—Enter the user name and password of a privileged Windows user in the domain where this MIIS instance is installed.
- **SQL Authentication**—Enter the user name and password of a valid SQL administrator.

## To configure the SQL server connection

1. In the Connect to Database page, specify the host name of the SQL Server instance where the OblixMA database is stored.

The SQL Server instance used by the Oblix Management Agent does not have to be the same as the one that MIIS uses.

2. In the Database field, enter OblixMA.
3. In the Table/View field, enter oblix\_main
4. In the Delta Table/View field, enter oblix\_delta.
5. Specify Windows or SQL authentication.
6. Enter the user name and password that you use for the SQL server or for Windows authentication, as appropriate.
7. Click Next.

The Configure Columns page appears. These columns refer to the tables created in SQL Server for the OblixMA database. There are two types of columns in this table:

- Columns prefixed with “ob” are control columns.
- The other columns contain identity data provisioned from a COREid workflow.

8. Continue with “Configuring the MIIS Database Columns” on page 564.

## Configuring the MIIS Database Columns

During this sequence, you will set the anchor (a column that uniquely identifies the row in a SQL table), add to the selected attributes list, identify the attribute where changes to information in the database are stored, identify keywords, and configure an optional disconnect rule.

### To configure the MIIS database columns

1. In the Configure Columns page, click the Set Anchor button.

The Set Anchor dialog appears. The anchor is a column that uniquely identifies the row in a SQL table.

2. Select the following:  
ob\_unique\_id
3. Click Add to move ob\_unique\_id to the Selected attributes list.
4. Click OK.

The Set Anchor dialog is dismissed and you return to the Configure Columns page.

5. Click Configure Delta.

The Configure Delta dialog appears. This identifies the attribute where changes to information in the database is stored.

6. In the Change type attribute drop-down list, select `ob_changetype`.

7. Specify the keywords that the OblixMA uses when recording actions in the `oblix_delta` table, as follows:

In the Modify field, enter MOD.

In the Add field, enter ADD.

In the Delete field, enter DEL.

Enter these keywords in capitals, as indicated above and click OK.

The Configure Delta dialog is dismissed and you return to the Configure Columns page.

8. Click Next.

The Configure Connector Filter page appears.

By default, you do not need to configure anything on this page. However, you may want to configure a disconnecter rule using the `ob_user_account_control` column. This column indicates status of the identity in the COREid System.

This column may have a value of `ACTIVATED` or `DEACTIVATED`.

Configuring the `ob_user_account` column ensures that if the user is deactivated in the COREid System, the user is also disconnected from the Metaverse.

9. Click Next.

The Configure Join and Projection Rules page appears. This page allows you to configure the rules by which data is coordinated between COREid and MIIS.

10. Continue with “Configuring Join and Projection Rules” on page 565.

## Configuring Join and Projection Rules

The Configure Join and Projection Rules page allows you to configure the rules by which data is coordinated between COREid and MIIS.

The list of data source attributes in the New Join Rule dialog is created during installation of the Oblix MA. The installation program generates an object template (.tpl) file based on attributes in the Metaverse. For more information on object templates, see the chapter on provisioning in the *NetPoint 7.0 Administration Guide, Volume 1*.

A join rule specifies the criteria by which an existing connector space object is linked to a Metaverse object. For example, the following join rule links objects in the connector space to matching objects in the Metaverse:

```
cs.user.cn=mv.person.cn
```

The join rule is based on the values of the attribute. The join occurs if the values of the attributes are the same.

A projection rule specifies that a connector space object should be linked to a newly created object in the Metaverse. A projection rule can link the OblixMA data to the Metaverse, for example, `cs.user -> mv.person`.

### To configure join and projection rules

1. On the Configure Join and Projection Rules dialog, click the New Join Rule button.

The New Join Rule dialog appears. For example, the following is a join rule that links the OblixMA connector space object to a Metaverse object based on the equality of the `cn` attribute:

```
cd.person#1cn->cn
```

2. In the New Join Rule dialog, select a data source attribute (or hold down the Control key to select more than one) that you will configure in a COREid workflow.

For example, you may want to add the `cn` and `sn` attributes to a join rule for a person.

3. Select a Mapping Type of Direct.
4. Select a Metaverse object type.

---

**Note:** You can join a connector space object to a new object based on matching values of the attribute as long as both classes contain the same attribute. For example, you can map the `cn` attribute to the person object class because `cn` is an attribute of that class.

---

5. Select a Metaverse attribute from the Metaverse attribute list box.
6. Click Add Condition.

---

**Note:** You can also create a new join rule for a different object class. For example, you can create a new join rule and use it to add the `cn` attribute to the Group object class.

---

7. When you finish creating join rules, click OK.

The Join Rule dialog is dismissed and you return to the Configure Join and Projection Rules dialog.

8. Click the New Projection Rule button.

The Projection dialog appears.

9. Select Rules Extension.

The OblixMA uses the rules extension to set the correct object type during projection.

10. Click OK.

11. Click Next.

The Configure Attribute Flow page appears. The attribute flow determines what attributes are written from the COREid to the Metaverse.

12. Continue with “Configuring the Attribute Flow” on page 567.

## Configuring the Attribute Flow

The attribute flow determines what attributes are written from the COREid to the Metaverse. The Configure Attribute Flow page enables you to configure the attributes for each identity that you are mapping from the Oblix connector space to the Metaverse.

### To configure the attribute flow

1. On the Configure Attribute Flow page, set the Mapping Type to Advanced to allow multi-selection of attributes.

For example, if you are mapping the cn from a COREid workflow to the Metaverse cn, your goal is to create an attribute flow similar to the following:

```
cd.person:<dn>,cn->mv.person.cn
```

The <DN> attribute type is a required component of any attribute flow. This attribute is used by the custom Oblix extension (ObMASynchronization) that works with the MA.

2. In the Data Source Object Type table, select the <DN> attribute type.

---

**Important:** You can select more than one attribute to be used for the source data, but the last source attribute in the rule is the one that is used to look up multi-valued attributes.

---

3. Select a flow direction of Import to allow importing data to the Metaverse from the OblixMA connector space.
4. Select the Metaverse object type and Metaverse attribute to which the data source attributes will be imported.
5. Click the New button.

The Advanced Import Attribute Flow Options dialog appears. This dialog shows a Rules Extension field with a flow rule that matches what you configured for the attribute flow.

6. On the Advanced Import Attribute Flow dialog, select Rules extension as the Mapping Type.

---

**Important:** Do *not* change the flow rule name. The OblixMA depends on the format of the default flow rule name.

---

For example, for the `cn` attribute of the person object, the flow rule name might be:

```
cd.person:<dn>,cn->mv.person.cn.
```

**cd**—The connected data source. In this case, it is the Oblix Management Agent for MIIS.

**person**—The object type of the data source object. This is ignored. The data source object type is determined from the value of the `ob_objectclass` column in the `oblix_main` table in the SQL server database.

**cn**—The source attribute. By convention, the right-most source attribute is the one that is referenced during lookup of multi-valued attributes.

**mv**—The Metaverse.

**person**—(Second instance in the example.) The object type of the Metaverse object.

**cn**—(Second instance in the example.) The target attribute.

7. Click OK.

The Advanced Import Attribute Flow dialog is dismissed.

8. Repeat this process for each attribute that you want to map.
9. When you finish, click Next.

The Configure Deprovisioning page appears.

---

**Note:** If you receive an error while configuring an attribute flow, be sure that the attributes being referenced do not have a DN reference type in MIIS. For example, if you want to map the groupmember attribute from COREid, this attribute cannot have a DN reference type in MIIS. You must configure the attribute as character data in MIIS.

---

10. Continue with “Configuring Deprovisioning” on page 569.

## Configuring Deprovisioning

On the Configure Deprovisioning page, you must select the box beside “Do not recall attributes contributed by objects from this management agent when disconnected.” to ensure that if you remove an identity from the Oblix MA, the attributes and values that have been written to the Metaverse remain in the Metaverse.

### To configure deprovisioning or MIIS

1. In the Configure Deprovisioning page, select the checkbox labeled “Do not recall attributes contributed by objects from this management agent when disconnected.”

---

**Important:** Selecting the checkbox ensures that if a row is deleted in the `oblix_main` table, the attributes and values that have been written to the Metaverse remain in the Metaverse.

---

2. Click Next.

The Configure Extensions page appears. This page allows you to select a management agent extension.

3. Continue with “Configuring Extensions” on page 569.

## Configuring Extensions

You complete the sequence by selecting the `obmasynchronization.dll`.

### To configure extensions

1. On the Configure Extensions page, click Select.

The Rules Extension Location appears.

2. Select `obmasynchronization.dll` and click OK.
3. Click Finish.
4. Continue with “Configuring Run Profiles for the Agent” on page 569.

## Configuring Run Profiles for the Agent

In order to execute the synchronization, run profiles need to be configured for the OblixMA. Typically, two types of run profiles are created:

- **One for Full Import**—This is typically run once to populate the connector space of the management agent.
- **One for Delta Import and Delta Synchronization**—This is run periodically, synchronizing only that data that has changed since that last synchronization.

### **To set up a Run Profile for Delta Synchronization**

1. Click the Management Agents button in the toolbar.
2. In the Actions list, click Configure Run Profiles.  
The Configure Run Profiles wizard starts up.
3. In the first page of the wizard, click “New Profile”.
4. In the Name field, enter “Delta Synchronization”.
5. Click Next.  
The Configure Step page is displayed.
6. In the Type field, select Delta Import and Delta Synchronization.
7. Click Next.  
The Management Agent Configuration page is displayed.
8. Click Finish.

### **To set up a Run Profile for Full Import**

1. Repeat the above procedure using the following input:  
**Name:** Full Import  
**Type:** Full Import (Stage Only).

---

**Note:** Continue with “Copying the MIIS Files to COREid” on page 570.

---

## **Copying the MIIS Files to COREid**

After you install the Oblix MA, configure the Oblix Management Agent in MIIS, and edit obmaconfig.xml, you need to copy three items to the COREid System installation directories so that the COREid System can interoperate with MIIS:

- The metadata file
- The object template file
- The license string

## To copy MIIS Files to COREid

1. Copy the file:

*Oblix\_MA\_install\_dir*\toCOREid\obmetadata.xml

where *Oblix\_MA\_install\_dir* is the directory where the Oblix Management Agent for MIIS is installed, for example, C:/Program Files/Oblix Management Agent

to

*COREid\_install\_dir*\oblix\config\obmetadata.xml

where *COREid\_install\_dir* is the directory where the COREid System is installed.

2. Copy the file:

*Oblix\_MA\_install\_dir*\toCOREid\miis.tpl

to

*COREid\_install\_dir*\oblix\config\template\miis.tpl

3. Copy the temporary license key can be found in the “value” parameter in the following file:

*Oblix\_MA\_install\_dir*\toCOREid\MIIS.LICENSE.tmp

Paste the license key into the COREid Connector for MIIS field in the COREid System Console, System Configuration > View Server Settings > Configure Licenses and click Save.

4. Restart the COREid Server to pick up these changes.

## Configuring the Database, COREid, and SSO

After configuring the Oblix Management Agent, you configure a license for the agent in the COREid System, and you configure an RDBMS profile to connect to the Oblix SQL Server database. This activity requires configuration in the Windows, COREid, and Access System administration consoles. You can only configure one profile for MIIS in the COREid administration console.

The following overview introduces the high-level steps you need to complete for this task and where to find more information.

### Task overview: Configuring the database, COREid, and SSO

1. Configure the MIIS license key in the COREid System Console, as described in “Configuring the SQL Database” on page 572.
2. Configure the ODBC and the OblixMA SQL Server database, as described in “Configuring the SQL Database” on page 572.

3. Configure a COREid workflow to send values for MIIS schema attributes to the Oblix MIIS Agent, as described in “Configuring a COREid Workflow for MIIS Provisioning” on page 573.
4. If the WebPass that the ObMA application communicates with is protected by a WebGate, you need to configure single sign-on between the MA and the WebPass, as described in “Configuring Single Sign-On” on page 576.

---

**Important:** In the following procedures, be sure that the DSN name specified in the COREid System matches the DSN name that you input in the Windows ODBC Data Source configuration tool.

---

## Configuring the SQL Database

You must configure ODBC in Windows Administrative Tools and configure the OblixMA SQL Server database in the COREid System.

### To configure ODBC for the SQL Server instance

1. From the Windows Start menu, select Administrative Tools > Data Sources > Windows 2003.
2. In the ODBC Data Source Administrator dialog, select System DSN.
3. Click Add.
4. In the Create a New Data Source dialog, select SQL Server.
5. Enter the following information:
  - Name**—enter the DSN name (for instance, Oblix MA).
  - Server**—The SQL Server that hosts the OblixMA database.
6. Select the authentication method that you chose when you installed the SQL Server.
7. Change the default database to OblixMA.

### To configure the SQL Server Profile in the COREid System

1. From the COREid System Console, select System Configuration > Configure Directory Options > Configure RDBMS Profiles > Add.
2. Supply the following information:
  - Name**—A logical name.
  - Used by**—MIIS.
3. In the Database Instance section of the page, click Add.  
The Create Database Instance page appears.

4. On the Create Database Instance page, supply the following information:

**Name**—A logical name.

**DSN name**—The DSN name you supplied in the ODBC Data Source Administration tool.

---

**Important:** The DSN name that you supply in the ODBC Data Source Administration tool must match that which you supply in the COREid System Console.

---

**Database name**—Optional.

**User name**—Optional if the Data Source uses Windows Authentication. If the Data Source uses SQL Server Authentication, provide the same user name that you used for authentication when configuring the Data Source.

**Password**—Optional if the Data Source uses Windows Authentication. If the Data Source uses SQL Server Authentication, provide the same password that you used for authentication when configuring the Data Source.

5. Click Save.
6. Save the profile.
7. Restart the COREid Server.

---

**Note:** You may want to configure multiple instances of the database for failover.

---

## Configuring a COREid Workflow for MIIS Provisioning

You must configure a COREid workflow to send values for MIIS schema attributes to the Oblix MIIS Agent. Configuring a COREid Workflow for MIIS provisioning includes the procedures below:

- “To configure an MIIS object class in the COREid System Console” on page 574
- “To configure the MIIS attributes you wish to use in a workflow” on page 574
- “To add an MIIS object class to a COREid application tab” on page 574
- “To add the MIIS attributes to a profile page panel” on page 575
- “To add the MIIS attribute to a workflow” on page 575
- “To test your OblixMA configuration” on page 576

---

**Important:** The following procedures assume prior experience with configuring objects and attributes in the COREid System Console. The *NetPoint 7.0 Administration Guide Volume 1* contains an in-depth discussion of configuring objects and attributes in the COREid System, and using these attributes in a workflow. If you are new to these topics, read the chapters on configuring objects, configuring COREid applications, and configuring workflows in this manual.

---

### **To configure an MIIS object class in the COREid System Console**

1. In the COREid System Console, select Common Configuration > Configure Objects > Add.
2. In the Schema Domain drop-down list, select MIIS.
3. In the Object Class drop-down list, select the object class that you want to add.
4. Select the class Type
5. Select Template as the object class Kind.
6. Select Auto Configure.
7. Click Save.

### **To configure the MIIS attributes you wish to use in a workflow**

1. In the COREid System Console, select Common Configuration > Configure Object Class and click the tab link.
2. Click Modify Attributes.
3. Select the first MIIS attribute that you wish to use in a workflow.
4. Give this attribute a display name that identifies it as an MIIS attribute.  
For example, you might give the attribute cn.person.miis a display name of MIIS full name.
5. Configure the MIIS password attribute (obattrbin.person.miis) and assign it a semantic type of Password.  
You must configure an attribute with a semantic type of password for password synchronization to work. Semantic types for Template objects are ignored with the exception of the Password semantic type.
6. Configure additional attributes as needed.

### **To add an MIIS object class to a COREid application tab**

1. In the COREid System Console, select User, Group, or Organization Manger Configuration > Configure Tab and click the tab link.
2. Click Modify.

3. In the Object Class(es) selection box, select the MIIS-related object that you wish to add to this tab.

You must associate an MIIS object class with a tab before associating attributes in this object class with a workflow.

4. Click Save.

### **To add the MIIS attributes to a profile page panel**

1. In the COREid System Console, Select User, Group, or Organization Manger Configuration > Configure Tab and click the tab link.
2. Click View Object Profile > Configure Panels.
3. Add the MIIS attributes to the desired panel.

You select attributes on this page according to the display name that you configured earlier. Attributes configured from `miis.tpl` need to be added to a profile panel so that Change Attribute workflows defined for those attributes can be triggered.

---

**Note:** Existing values for attributes from `miis.tpl` are not shown. (This is considered as part of Inbound Provisioning from MIIS, a feature not available in the current release.)

---

### **To add the MIIS attribute to a workflow**

1. From the User, Group, or Organization Manager, select Configuration > Workflow Definition.
2. Click New.
3. Provide a workflow name that indicates the workflow is for MIIS provisioning, and select the workflow type and domain.
4. Click Next.
5. If this is a create workflow type, select a target.
6. Click Next.
7. Select the attributes that you wish to use for this workflow.  
For example, for a create user workflow you might select `cn.person.miis`, and `password.person.miis`.
8. Select a step participant.
9. Save the step.

10. Configure other steps of this workflow, as needed.

When you configure the Commit step of this workflow, be sure that MIIS-specific data is saved to the MIIS domain. Commit steps for the MIIS schema domain are routed through the MIIS agent that is installed with the COREid System.

11. Configure a separate Commit step for each schema domain to which you want to save data.

The workflow is resumed by the ObMA application through IDXML. The status will either be success, if all passwords are successfully set for all Management Agents configured for password synchronization, or failure, if a password cannot be set for at least one Management Agent. The workflow can be configured so that an Error Report step is generated if the commit step to MIIS resumes with a status of failure. The actor comments for that page will list the Management Agent and the last password set status for each Management Agent for which password set failed. In addition, using Retry, the workflow can be configured to cycle back to a previous step, collect a new password and attempt to re-synchronize the password.

### **To test your OblixMA configuration**

1. Log in as a user to the application that uses the workflow that you have created.
2. Enter data for the workflow.
3. If necessary, log in under an identity required to approve the workflow.

Workflows that commit passwords to MIIS are placed in a Pending state until password synchronization is complete.

4. View the data in the OblixMA SQL Server database, or view the data in the target MIIS application to see if the data has been provisioned correctly.

## **Configuring Single Sign-On**

If the WebPass that the ObMA application communicates with is protected by a WebGate, you need to configure single sign-on between the MA and the WebPass.

---

**Note:** The following procedure includes steps for configuring an AccessGate. The AccessGate configuration specified in the following procedure uses an AccessGate plug-in that was automatically installed with the NetPoint 7.0 Access System.

---

## To configure single sign-on

1. Set the parameter `wp_sso` to true in `obmaconfig.xml`.
2. Create an AccessGate profile in the Access Manager.  
See the *NetPoint Administration Guide Volume 2* for details.
3. Configure the new AccessGate using the `configureAccessGate` tool located in:  
`OblixMA_install_dir/AccessServerSDK/tools/configureAccessGate`  
where `OblixMA_install_dir` is the directory where the Oblix Management Agent for MIIS is installed.  
  
For more information on the `configureAccessGate` tool, see the *NetPoint Administration Guide Volume 2*.
4. Ensure that the following parameters used by the ASDK to generate the SSO cookie have correct definitions in `obmaconfig.xml`:
  - `wp_sso_cookie_url`
  - `wp_sso_cookie_path`
  - `wp_sso_cookie_domain`
  - `wp_sso_cookie_ipaddress`

---

**Note:** In most cases, the default values for these parameters can be used.

---

See “Customize the Management Agent Configuration File” on page 577 for details.

## Customize the Management Agent Configuration File

This file is read by the ObMA application and the Management Agent and Metaverse extensions. You may need to edit this file, depending on your MIIS configuration or business requirements. Reasons include:

- Enforcing secure connections between components.
- Enabling single sign-on.
- Restricting password synchronization to a subset of available Management Agents.
- Setting the retry count for password synchronization.
- Setting the log level for the ObMA application and the OblixMA Management Agent and Metaverse extensions, `obmasynchronization.dll` and `obmv synchronization.dll` respectively.

This section discusses the configurable parameters for the ObMA application.

This file is located in

```
Oblix_Management_Agent_install_dir\oblix\config\obmaconfig.xml
```

where *Oblix\_Management\_Agent\_install\_dir* is the directory where the Oblix Management Agent is installed.

## Password Synchronization

The file contains the following section on password synchronization:

```
<oblix-section name="OblixPasswordSynchronization">
  <oblix-param name="obpwd_maxretry">
    <oblix-value>3</oblix-value>
  </oblix-param>
  <oblix-param name="obpwd_ma" />
</oblix-section>
```

In the `OblixPasswordSynchronization` section, the parameter `obpwd_ma` contains a list of Management Agent names on which to attempt a password set operation. By default, all Management Agents are used. To specify individual Management Agents, list the individual names, as follows:

```
<oblix-param name="obpwd_ma">
  <oblix-value>Name of Management Agent 1</oblix-value>
  <oblix-value>Name of Management Agent 2</oblix-value>
  <oblix-value>Name of Management Agent 3</oblix-value>
</oblix-param>
```

## WebPass Configuration

The following parameters affect the configuration of the WebPass in the Management Agent Configuration file:

**wp\_ssl**—Determines whether HTTPS is used for IdentityXML requests that are made during password synchronization. If set to true, the ObMA application tries to establish a secure connection with WebPass during password synchronization.

**wp\_trusted\_cert**—Set this parameter to true to force the certificate issued by the Web server to be trusted. This flag is only used if `wp_ssl` is also set to true.

An example of the WebPass configuration section of this file:

```
<oblix-section name="WebPass">
  ...
  <oblix-param name="wp_ssl">
    <oblix-value>true|false</oblix-value>
  </oblix-param>
  <oblix-param name="wp_sso">
```

```

        <obl ix-value>true|false</obl ix-value>
    </obl ix-param>
    <obl ix-param name="wp_trusted_cert">
        <obl ix-value>true|false</obl ix-value>
    </obl ix-param>
</obl ix-section>

```

**wp\_sso**—Determines whether an IdentityXML request that is sent to a WebPass protected by a WebGate attempts to generate an ObSSOCookie during this process. This flag makes use of the Access Server SDK that was installed with the Oblix Management Agent for MIIS.

**wp\_sso\_cookie\_url**—The ObSSOCookie URL.

**wp\_sso\_cookie\_path**—The ObSSOCookie path, for example /identity/obl ix.

**wp\_sso\_cookie\_domain**—The domain of the Web server that hosts WebPass. The default is the domain of the local host, for example, .obl ix.net. Use the fully qualified domain name and note that the leading “dot” is required.

**wp\_sso\_cookie\_ipaddress**—The IP address against which the WebGate will do an IP validation check if IP validation is enabled. See the information on the IPValidation parameter in the WebGateStatic.lst in the *NetPoint 7.0 Administration Guide, Vol. 2*. The default is no IP address.

## MIIS Configuration

The MIIS section of obmaconfig.xml affects MIIS operation. Note that the values for the parameters in this section, for the most part, remain constant. Only the connection strings could be subject to change. For example, you may want to connect using different credentials than those supplied during installation.

For more information on other parameters in this section of the configuration file, see the Microsoft documentation:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfsystemdatasqlclientsqlconnectionclassconnectionstringtopic.asp>

The following parameter is of interest:

- **miis\_sql\_constr**—This creates a connection string to the database for the Oblix Management Agent. It is an ADO.NET connection string.

## WMI Configuration

The following parameter is of interest:

- **wmi\_path**—This is the path to WMI.

# Configuring Logging

There are three files that are used for logging:

- **obma.log**—The ObMA application writes logs to this file.
- **oblog.log**—The installer uses this file to capture installation information. For information on the format and configuration of the oblog.log file, see the chapter on logging, auditing, and SNMP in the *NetPoint 7.0 Administration Guide, Vol. 1*.
- **obmaextension.log**—The ObMASynchronization and ObMVSynchronization extensions log to this file, which by default is located in  
C:\Program Files\Microsoft Identity Integration Server\MaData.

## Logging Parameters

The file obmaconfig.xml discussed in “Customize the Management Agent Configuration File” on page 577 contains a few parameters that control logging:

- **obma\_extension\_loglevel**—Controls the log level below which logging will be done by the ObMASynchronization and ObMVSynchronization extensions. Possible values are 0 (fatal), 1 (Error), and 2 (Info). The default is 1 (Info). Example:

```
<oblix-param name="obma_extension_loglevel">  
  <oblix-value>2</oblix-value>  
</oblix-param>
```

- **filename**—The name of the log file used by ObMASynchronization and ObMVSynchronization. The default is obmaextension.log. Example:

```
<oblix-param name="obma_extension_logfile">  
  <oblix-value>obmaextension.log</oblix-value>  
</oblix-param>
```

MIIS allows you to configure all Management Agents to write to one log file or to different log files. This controlled by setting a parameter in the following file:

```
C:\Program Files\Microsoft Identity Integration  
Server\Extensions\logging.xml
```

The parameter use-single-log determines if logs are written to one or more files. A value of true directs logging from all Management Agent extensions to one file, including logging from the OblixMA extensions. Example:

```
<rules-extension-properties>  
  <logging>  
    <use-single-log>>false</use-single-log>
```

```
<file-name>MIIS.log</file-name>
<logging-level>2</logging-level>
</logging>
</rules-extension-properties>
```

## About the ObMA Application

The ObMA application is used to run profiles of the OblixMA, synchronize passwords, and for data clean-up or deleting rows from the OblixMA database tables. A row in the `oblix_main` table represents an identity provisioned by a COREid workflow. Installing the COREid Connector for MIIS will install two scheduled tasks that periodically run the ObMA application, as described in “Confirming Installation” on page 559.

As discussed under “Preparing for Password Synchronization” on page 557, the ObMA application uses the WMI Provider for MIIS to set passwords on target connector space objects. Only members of the `MIISPasswordSet` and the `MIISAdmin` security groups can perform this function.

This means that the “RunAs” user for the ObMA application must be a member of these two groups. In addition, the ObMA application executes SQL statements against the OblixMA database. The “RunAs” user must also have permission to execute queries and updates to the OblixMA database.

In case of the `-d` and `-dd` options discussed in “Command-Line Parameters” on page 581, the ObMA application queries the `mms_connectorspace` table of the MIIS database. If these options are used, the “RunAs” user must have permission to execute select statements on this table. Note that the ObMA application does not modify the data.

## Command-Line Parameters

Usage

```
[-r { run_profile_name[,...n] } ] | [-p] | [-d | -dd {
'oblix_delta' | 'oblix_pwd' | 'oblix_main' }]
```

The functionality is always executed in the following sequence regardless of the order of the parameters on the command-line:

```
-r -p -d oblix_delta -d oblix_main -d oblix_pwd
```

where

**-r**—executes one or more Run Profiles of the OblixMA Management Agent. The Run Profile names are provided as a comma separated list. Names with spaces must be enclosed in double quotes.

**-p**—executes password synchronization

**-d**—executes data deletion. The name of the table is either `oblix_main`, `oblix_delta` or `oblix_pwd`. Using this option, rows will be removed from the specified table given the listed pre-condition:

**ooblix\_main**—the row has been synchronized to the Metaverse. Deleting a row in `oblix_main` will remove referencing rows in `oblix_mv`, the multi-valued attribute table, and `oblix_pwd`; for the latter only if password synchronization has occurred.

**ooblix\_delta**—the row has been staged to the connector space or the row is orphaned (there is no corresponding row in `oblix_main`)

**ooblix\_pwd**—the password referenced by the row has been synchronized

**-dd**—executes data deletion, but with less stringent pre-conditions than `-d`. The name of the table is either `oblix_main` or `oblix_pwd`. Using this option, rows will be removed from the specified table given the listed pre-condition.

**ooblix\_main**—the row has been synchronized to the Metaverse or the row is a disconnecter. Deleting a row in `oblix_main` will remove referencing rows in `oblix_mv`, the multi-valued attribute table, and `oblix_pwd`; for the latter only if password synchronization has occurred.

**ooblix\_pwd**—the row is deleted even if password synchronization has not yet run.

## Examples

**-r "Delta Synchronization" -p -d oblix\_delta**—Typical use case that does a delta synchronization, password synchronization and delta table clean-up.

**-d oblix\_main**—Removes all identities from `oblix_main` that have no unsynchronized data.

**-dd oblix\_main -dd oblix\_pwd**—Removes all identities that have no unsynchronized data, as well as disconnected identities as well as all rows in `oblix_pwd`.

## Using a COREid Workflow to Delete Identities from Metaverse

This section describes the additional configuration steps required so that a COREid workflow can be used as the authoritative source for deleting identities from the Metaverse, which in turn can start the deprovisioning or disconnection process in various target data stores.

## **Task overview: Using a COREid workflow to delete identities from the Metaverse**

1. Define a workflow with a delete step with MIIS as the schema domain.
2. With Identity Manager, in Metaverse Designer, set the Object Deletion Rule to be Rules Extension.
3. Integrate obmvsynchronization.dll with your Metaverse extension.
  - a) In your Metaverse extension, load obmvsynchronization.dll. To see how to dynamically load other assemblies and dynamically find implementations of the IMV Synchronization interface in those assemblies, see the following code example from the MIIS Developer Reference—Example: Calling Other Assemblies from a Rules Extension.
  - b) In the implementation of your method ShouldDeleteFromMV, call the same method on ObMVSynchronization (this is the OblixMA implementation of the IMV Synchronization interface).
  - c) A return value of true indicates that COREid workflow has provisioned a delete for this identity; false means that the deletion originated from some other Management Agent or that the deletion originated from the OblixMA Management Agent, but only because an identity was removed using the -d option of ObMA, not because it was provisioned by COREid workflow.
  - d) Based on this value, action can be taken in your Metaverse extension to initiate deprovisioning or disconnection of identities in various target data stores.



# 15 Integrating NetPoint SSO with SharePoint Portal Server

This chapter explains how to integrate NetPoint 7.0 with the Microsoft SharePoint Portal Server (SPPS) 2003. It covers the following topics:

- “About NetPoint and the SharePoint Portal Server” on page 585
- “Supported Platforms and Requirements” on page 586
- “Request Processing by the SPPS Integration” on page 588
- “Integrating NetPoint with SPPS” on page 589
- “Setting Up Impersonation” on page 595
- “Completing NetPoint to SPPS Integration” on page 604

## About NetPoint and the SharePoint Portal Server

NetPoint provides a full range of identity management and security functions, including: Web-based single sign-on (SSO), user self-service and self-registration, user provisioning, reporting and auditing, policy management, dynamic groups, and delegated administration. NetPoint integrates with all leading directory servers, application servers, Web servers, and enterprise applications.

SharePoint Portal Server (SPPS) is a secure, scalable, enterprise portal server that builds on Windows Server 2003 Microsoft Internet Information Services (IIS) and Windows SharePoint Services (WSS). SPPS can aggregate SharePoint sites, information, and applications into a single, easy-to-use portal. In addition to WSS functionality, SPPS incorporates additional features such as News and Topics as well as personal and public views for My Site, and so on.

Once NetPoint has been integrated with SPPS, NetPoint handles user authentication through an ISAPI filter for IIS and an ISAPI wild card extension, which enables SSO between NetPoint and SPPS. WSS handles resource request authorization for all SPPS resources.

Such integration enables authenticated users to enjoy SSO access not just to SPPS resources, but also NetPoint-protected resources, which can reside on the full range of NetPoint-supported platforms (such as Windows, Solaris, or Linux,) or application servers (such as WebLogic or WebSphere).

## About Windows Impersonation

NetPoint/SPPS integration utilizes the Windows impersonation feature, which enables a trusted user in the Windows server domain to assume the identity of any user requesting an SPPS target resource. This trusted impersonator maintains the identity context of the user while accessing the resource on behalf of the user.

Impersonation is transparent to the user; access appears to take place directly, as if the SPPS resource were a resource within the NetPoint domain.

## Supported Platforms and Requirements

Successful NetPoint integration with SPPS requires both NetPoint and Microsoft components, which must be installed and configured to support impersonation as well as integration. See the following topics for details:

- “Required Microsoft Components” on page 586
- “Required NetPoint Components” on page 588

## Required Microsoft Components

Table 40 lists the Microsoft components required to integrate NetPoint with SPPS.

**Table 40** Microsoft Requirements

Component/Feature	Requirement/Comment
Operating System	Windows Server 2003 for the SPPS host. <b>Note:</b> Any of the five editions is acceptable. <b>Also:</b> The Active Directory Domain Controller must reside on a Windows Server 2003 machine, although that machine doesn't have to be the one hosting SPPS.
Extended Services	<ul style="list-style-type: none"><li>• Internet Information Services (IIS) 6.0. Note: You must install IIS on the machine that will host SPPS after installing Windows Server 2003 on that machine.</li><li>• Windows SharePoint Services (WSS) 2.0. These services install automatically when you install the SharePoint Portal Server. (See Portal Server in the second row following).</li></ul>

**Table 40** Microsoft Requirements

Component/Feature	Requirement/Comment
Directory Service	<p>Active Directory. You install this after installing Windows Server 2003. You can connect SPPS directly to the Active Directory Domain Controller or to a different instance of Active Directory.</p> <ul style="list-style-type: none"><li>• The Active Directory Domain Controller can reside on the same machine as your SPPS installation. If it does, however, SPPS requires an instance of SQL Server (not Desktop SQL) installed on a machine within the Active Directory domain.</li><li>• Alternatively, you can connect SPPS to an Active Directory Domain Controller residing on a different Windows Server 2003 machine.</li><li>• Another option is to connect SPPS to a non-domain controller instance of Active Directory, which can reside on the machine hosting SPPS or on any other machine in your Active Directory domain.</li></ul> <p><b>Note:</b> For all scenarios mentioned here, SPPS can use either Desktop SQL or an instance of SQL Server installed on a machine within the Active Directory domain.</p>
Portal Server	SharePoint Portal Server (SPPS). After installing Active Directory, you install SPPS on a machine where Windows Server 2003 and IIS are already installed.
Security Service	Kerberos Key Distribution Center (KDC) installs automatically as part of Windows Server 2003.

## Required NetPoint Components

The NetPoint components in Table 41 are required to integrate NetPoint with SPPS.

**Table 41** NetPoint Requirements

Item	Requirement/Description
WebGate	<p>The ISAPI version WebGate that ships with NetPoint 7.0 must reside on the same machine as SPPS.</p> <p>Within the context of NetPoint/SPPS integration, this WebGate is an ISAPI filter that intercepts HTTP requests for Web resources and works with the Access Server to authenticate the user who made the request. If authentication is successful, the WebGate creates an ObSSO cookie and sends it to the user's browser, thus facilitating SSO. The WebGate also sets impersonate as a HeaderVar action for this user session.</p>
IISImpersonationExtension.dll	<p>This NetPoint 7.0 dll is an IIS wildcard extension that checks whether the Authorization Success Action HeaderVar has been set to impersonate, and if it has been, the dll creates a Kerberos U4S2Self ticket so that the special trusted user in the SPPS Active Directory can impersonate the user who originally made the request.</p> <p>When you run the NetPoint 7.0 ISAPI WebGate installation wizard, IISImpersonationExtension.dll installs automatically within the WebGate directory structure, but you still need to configure the dll manually to enable impersonation and SPPS integration.</p>
NetPoint Directory	<p>NetPoint can be connected to any supported directory service including, but not limited to, LDAP and Active Directory. It can even connect to the same instance of Active Directory used by SPPS.</p> <p>In any case, the NetPoint Directory does not have to reside on the same machine as SPPS and the WebGate protecting it.</p>
Other NetPoint Components	<p>NetPoint/SPPS integration also requires installation of the other standard NetPoint system components such as the Access Server with which the WebGate protecting your SPPS installation is configured to interoperate. For details see the <i>NetPoint Installation Guide</i>.</p> <p>Except for the WebGate protecting SPPS, your NetPoint components do not need to reside on the machine hosting SPPS.</p> <p><b>Note</b> that if you install either Access Manager or WebPass (or both) on the same IIS virtual server as SPPS, you must exclude the URL paths to those components through SharePoint Managed Paths. (See "To define managed paths in SharePoint" on page 594). You may find it easier to install Access Manager and WebPass on a machine other than the one on which SPPS resides or, at the very least, on an IIS virtual server other than the one on which SPPS has been installed.</p>

## Request Processing by the SPPS Integration

NetPoint uses the Windows impersonation feature to facilitate user access to SPPS

resources.

### **Process overview: Request processing with the SPPS integration**

1. The user requests access to an SPPS resource.
2. The WebGate ISAPI filter protecting SPPS intercepts the request, determines whether the target resource is protected, and if it is, challenges the user for authentication credentials.
3. If the user supplies credentials and the Access Server validates them, the WebGate sets an ObSSO cookie in the user's browser, thus enabling SSO.

The WebGate also sets an HTTP header variable called "impersonate", whose value is set to the authenticated user's LDAP uid (or samaccountname, if the user account exists in Active Directory, or userPrincipalName, if the user account exists in a multi-domain Active Directory forest).

---

**Note:** At this point, IIS considers the user to be anonymous, since the impersonation has not yet been set.

---

4. The NetPoint ISAPI wildcard extension IISImpersonationExtension.dll checks for the Authorization Success Action header variable named impersonate.

When such a header variable exists, the wildcard extension obtains a Kerberos ticket for the user. This Service for User to Self (S4U2Self) impersonation token enables the designated trusted user to assume the identity of the requesting user and obtain access to the target resource through IIS and SPPS.

## **Integrating NetPoint with SPPS**

There are several phases to integrate NetPoint with the SharePoint Portal Server.

### **Task overview: Integrating NetPoint with SPPS includes**

1. "Installing Microsoft Components" on page 589
2. "Installing NetPoint Components" on page 593
3. "Setting Up Impersonation" on page 595
4. "Completing NetPoint to SPPS Integration" on page 604

## **Installing Microsoft Components**

Except where noted, all Microsoft SPPS-related components must be installed on the same host machine, including the following software:

- Windows Server 2003

- Microsoft IIS v6 Web Server
- SPPS (and underlying WSS)

The following Microsoft components can be installed on machines other than the one hosting the main SPPS installation:

- Active Directory (see Table 40 on page 586 for installation location details).
- SQL Server must be installed on a machine in the Active Directory domain only if the Active Directory Domain Controller is installed on the same machine as SPPS. See Table 40 on page 586 for installation location details.
- Additional SPPS frontend servers
- One or more backend servers containing SPPS resources such as Web pages, documents, or applications

---

**Note:** All of the machines hosting the preceding SPPS-related components must be in the same Active Server domain as the SPPS server you are installing.

---

The following task overview includes references to documentation that provides procedures and steps you need to complete when installing the Microsoft components for this integration.

### **Task overview: Installing Microsoft Components**

1. Install Windows Server 2003 on the machine that will host your SPPS installation, as described in the appropriate Microsoft documentation.
2. Install IIS on the machine that will host your SPPS installation, as described in the appropriate Microsoft documentation.
3. Install Active Directory, as described in the appropriate Microsoft documentation and the *NetPoint 7.0 Installation Guide*; also, see Table 40 for installation location considerations.
4. If SPPS and Active Directory Domain Controller reside on the same machine, you must also install Microsoft SQL Server on that machine as well, as described in Table 40.
5. Install SharePoint Services and SharePoint Portal Server on the root virtual server (which uses port 80, by default) of your IIS installation or on some other IIS virtual server.
6. Create and set up your portal
7. After you install SPPS, stop and test the installation to ensure it operates correctly before you integrate with NetPoint.

## Task Overview: Creating and setting up a portal

1. Create a portal
2. Upload a test document
3. Create audiences
4. Edit audiences (if necessary)
5. Compile audiences

### To create a portal

1. In the Portal Site and Virtual Server Configuration section of the SharePoint Portal Server Central Administration page for server on which you wish to create a portal, click Create a portal site.
2. In the Portal Creation Options section, click Create a portal.
3. In the Site Name section, in the Name box, type a name for the portal site. This name will appear at the top of most pages for the portal site.
4. In the Virtual server list within the Site URL section, click the existing virtual server on the server that will host the portal site.
5. In the URL box, type the URL through which users connect to the portal site. By default, this URL is `http://server_name/`. If you choose a virtual server that has a port number other than 80, the port number appears as part of the URL, e.g., `http://server_name:port_number/`

---

**Note:** Note: Make e sure to specify the load-balanced URL, not the local server URL.

---

6. In the Account name box of the Owner section, type the account name of the portal site owner in the format `Domain\user_name`. The portal site owner manages content and user access.
7. In the E-mail address box, type the e-mail address for the portal site owner, then click OK.
8. On the Create Portal Confirmation for `server_name` page, click OK to begin creating the portal site.
9. The Operation Status page displays as the portal is created. Following successful portal site creation, the Operation Successful page appears. At this point, you can begin detailed configuration of the portal site.

### To upload a document to the portal

1. Using your Web browser, navigate to the home page for the portal.
2. Select Upload Document from the Actions list.

3. On the Upload Document page, click Browse, navigate to the document you wish to add, then click Open.

To add multiple documents simultaneously, click Upload Multiple Files.

To replace a file of the same name within the library, select the checkbox titled “Overwrite existing file(s)”

4. Click Save, then click Close.

### **To create audiences**

1. Audiences, which are based on jobs or tasks within an organization, match specified users to target content while preventing all other users from viewing that content. On the Managing Audiences page for the site you wish to configure, click Create audience.
2. On the Create Audience page, type a name and description for the audience.
3. Click either “Satisfy all rules” or “Satisfy any of the rules,” then click OK.
4. After the Add Audience Rule page appears, add whatever rules you wish to govern access to the site content. (You can also add rules through the View Audience Properties page). For details, consult the Microsoft SPPS documentation on Adding and Editing Audience Rules.
5. Compile the audience so that the content is targeted to that audience. See “To compile audiences” on page 592.

### **To edit audiences**

1. On the View Audience Properties page for the site you are configuring, click View Audience Properties, then click Edit audience.
2. On the Edit Audience page, change the name or description of the audience, as necessary.
3. Click either “Satisfy all rules” or “Satisfy any of the rules,” then click OK.
4. When the View Audience Properties page reappears, Add, Delete, or Edit the audience rules, as necessary.
5. Review the statistics for the audience, checking the number of current members and the most recent time of compilation. When you are satisfied with all the settings for the audience and the rules associated with that audience, compile the audience so that your changes take effect. See “To compile audiences” on page 592.

### **To compile audiences**

1. Any changes you make to an audience or the rules associated with them do not take effect until you compile the audience. Navigate to the Manage Audiences page and check the compilation status and most recent compilation time for the

audience you wish to compile. (You can also view the number of incomplete audiences on this page).

2. Either start a compilation or set a compilation schedule.

## Installing NetPoint Components

The ISAPI Webgate for SPPS must be installed on the same machine as SPPS. The rest of your NetPoint installation can reside on the same machine or any other machine.

If you choose to install NetPoint on a different machine (which can be a Solaris, Linux, or Windows machine), it can be set up for Active Directory (if the host machine runs Windows Server 2003) or some other directory service, such as NetScape Directory Server (if the machine runs Solaris or Linux, for example).

If both NetPoint and SPPS are set up for different instances of Active Directory, both instances must belong to the same Active Directory domain.

### To install NetPoint Components for SPPS integration

1. On either the same machine that hosts SPPS (or on a different machine), install a COREid Server and a WebPass, then set up the COREid System as described in the *NetPoint 7.0 Installation Guide* and see Table 41 on page 588 for WebPass installation considerations.
2. On either the same machine that hosts SPPS (or a different machine), install Access Manager and one or more instances of the Access Server, as described in the *NetPoint 7.0 Installation Guide* and Table 41 on page 588.
3. On the machine hosting the SPPS instance you are trying to integrate, install a NetPoint 7.0 ISAPI WebGate.

The `IISImpersonationExtension.dll` will be installed as part of the package in the following directory:

```
WebGate_install_dir\access\Obliv\apps\webgate\bin\
```

where *WebGate\_install\_dir* is the directory where you installed the WebGate.

4. If you installed Access Manager or WebPass on the same IIS virtual server as SPPS, complete activities in “Defining Managed Paths in SharePoint” on page 594.

## Defining Managed Paths in SharePoint

You complete the following procedure only if the Access Manager or WebPass resides on the same IIS virtual server as SPPS *and* listens to the same port as that IIS virtual server. For instance, the default virtual IIS server uses port 80, as do many Access Manager and WebPass installations; therefore, you need to change the port used by one application or exclude the path used by the NetPoint component through the Define Managed Paths feature in SharePoint.

### To define managed paths in SharePoint

1. Select Start > Administrative Tools > SharePoint Central Administration.
2. In the Virtual Server Configuration section, click Configure virtual server settings.
3. In the Virtual Server list, click Default Web Site or the name of the virtual server on which both SPPS and the NetPoint components are installed.
4. In the Virtual Server Management section, select Define Managed Paths.
5. In the Add a Path section, type the path to Access Manager or WebPass, then click the button marked Excluded path.
6. Click OK to add the path to the list of excluded paths.

**Figure 18** Defining Managed Paths in SharePoint

Windows SharePoint Services  
**Define Managed Paths**

---

Use this page to specify which paths in the URL namespace are managed by Windows SharePoint Services.

---

**Current Virtual Server**

Note the current virtual server name. To change virtual servers, go to the [Choose Virtual Server](#) page.

Virtual Server Name:	robinsportalserver
URL:	http://venice:82/
Version:	6.0.2.5530

---

**Included Paths**

This list specifies which paths within the URL namespace are managed by Windows SharePoint Services.

[Remove selected paths](#)

Path	Type
<input type="checkbox"/> (root)	Explicit inclusion
<input type="checkbox"/> sites	Wildcard inclusion
<input type="checkbox"/> personal	Wildcard inclusion

---

**Excluded Paths**

This list specifies which paths within the URL namespace are not managed by Windows SharePoint Services. Excluded paths take precedence over included paths.

[Remove selected paths](#)

Path
<input type="checkbox"/> uddipublic
<input type="checkbox"/> uddi

---

**Add a New Path**

Specify the path within the URL namespace to include or exclude. You can include an exact path, or all paths subordinate to the specified path.

Use the **Check URL** button to ensure that the path you include or exclude is not already in use for existing sites or folders, which will open a new browser window with that URL.

Path: \*

  
Note: To indicate the root path for this virtual server, type a slash (/).

Type:

Excluded path

Included path

Type:

## Setting Up Impersonation

Setting up impersonation, whether for SPPS integration or for use by some other application, is described in the following sections:

### Task overview: Setting up impersonation

1. Create a trusted user account for only impersonation in the Active Directory connected to SPPS, as described in “Creating a Trusted User Account” on page 596.
2. Give the trusted user the special right to act as part of the operating system., as described in “Assigning Rights to the Trusted User” on page 597.
3. Bind the trusted user to the WebGate by supplying the authentication credentials for the trusted user, as described in “Binding the Trusted User to Your WebGate” on page 598.

4. Add a header variable named impersonate to Authorization Success Action in the policy domain for impersonation, as described in “Adding an Impersonation Action to a Policy Domain” on page 599.
5. Configure IIS by adding the IISImpersonationExtension.dll to your IIS configuration, as described in “Adding an Impersonation dll to IIS” on page 600.
6. Test impersonation, as described in “Testing Impersonation” on page 601.

## Creating a Trusted User Account

This special user should not be used for anything other than impersonation.

### To create a trusted user account

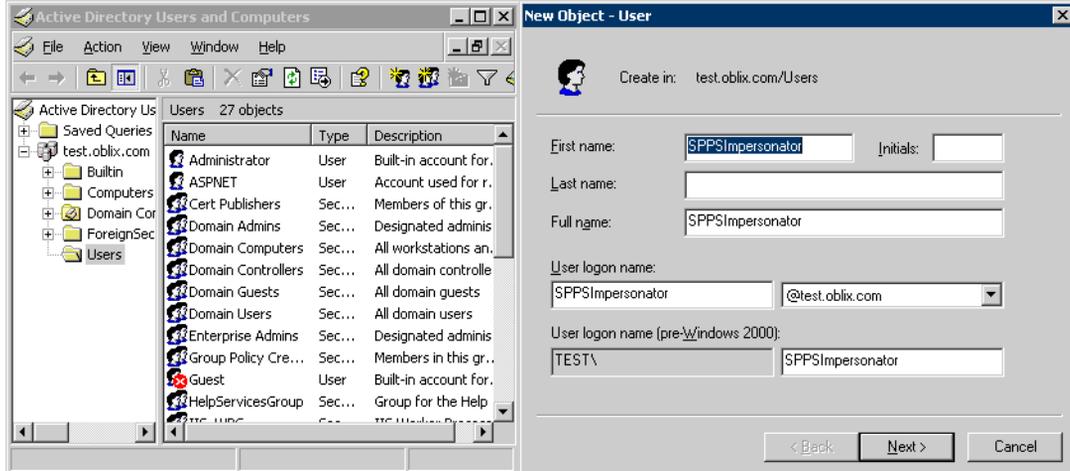
1. On the Windows 2003 machine hosting your SPPS installation, select Start > Programs > Manage Your Server > Domain Controller (Active Directory) > Manage Users and Computers in Active Directory.
2. In the Active Directory Users and Computers window, right-click Users on the tree in the left pane, then select New > User.
3. In the First name field of the pane entitled New Object - User, enter an easy-to-remember name such as *SPPSImpersonator*.
4. Copy this same string to the User logon name field, then click Next.
5. In succeeding panels, you will be asked to choose a password and then retype it to confirm.

---

**Note:** Oblix recommends that you chose a very complex password, because your trusted user is being given very powerful permissions. Also, be sure to check the box marked Password Never Expires. Since the impersonation extension should be the only entity that ever sees the trusted user account, it would be very difficult for an outside agency to discover that the password has expired.

---

**Figure 19** Setting up a Trusted User Account for Windows Impersonation



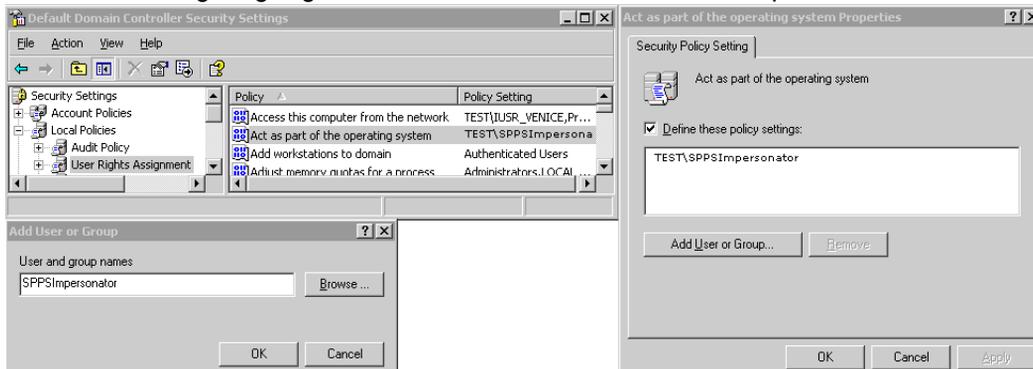
## Assigning Rights to the Trusted User

You need to give the trusted user the right to act as part of the operating system.

### To give appropriate rights to the trusted user

1. Select: Control Panel > Administrative Tools > Domain Controller Security Policy.
2. On the tree in the left pane, click the plus icon (+) next to Local Policies.
3. Click User Rights Assignment on the tree in the left pane.
4. Double-click “Act as part of the operating system” in the right pane.
5. Click Add User or Group.
6. In the Add User or Group panel, type the User logon name of the trusted user (SPPSImpersonator in our example) in the User and group names text entry box, then click OK to register the change.

**Figure 20** Configuring Rights for the Trusted User in Windows Impersonation



## Binding the Trusted User to Your WebGate

You need to bind the trusted user to the WebGate by supplying the authentication credentials for the trusted user, as described below.

### To bind your trusted user to your WebGate

1. Point your browser to your Access System Console. For example:

`http://hostname.domain.com:port/access/oblix`

where *hostname* is the DNS name of the machine hosting your Access Manager, *domain* is the name of the server domain to which the machine belongs, and *port* is the number of the port to which Access Manager listens.

2. Navigate to Access System Console > Access System Configuration > AccessGate Configuration.

3. Select the name of the Webgate you want to modify.

The Details for NetPoint AccessGate page appears with a summary of the configuration information for this WebGate. At the bottom of this Web page are fields for Impersonation Username and Impersonation Password.

4. Click the Modify button at the bottom of the Details for NetPoint AccessGate page.
5. After the Modify NetPoint AccessGate page appears, scroll to the bottom and enter the username and password for the trusted user account you created through the task on page 596.

For example:

Impersonation username:	<input type="text" value="SPPSImpersonator"/>
Impersonation password:	<input type="password" value="*****"/>
Re-type impersonation password:	<input type="password" value="*****"/>

6. Click the Save button to commit the changes and return to the Details page.

A bind has been created for the WebGate and the trusted user. The WebGate is now ready to provide impersonation on demand. The demand is created by an Authorization Success Action in a policy domain created for impersonation.

## Adding an Impersonation Action to a Policy Domain

You must create or configure a NetPoint policy domain to protect your SharePoint resources. You do this by adding an Authorization Success Action with a return type of “headerVar,” the “name” parameter set to “Impersonate”, and the “return attribute” parameter set to “samaccountname” for a single-domain Active Directory installation or “userPrincipalName” for a multi-domain Active Directory forest.

You must also choose an easy-to-remember name for the domain, such as *Impersonation Policy Domain*.

For details on creating a NetPoint policy domain, see the *NetPoint 7.0 Administration Guide Volume 2*.

### To add an impersonation action to your policy domain

1. Navigate to the Access System Console and log in. For example:

`http://hostname.domain.com:port/access/oblix`

where *hostname* is the DNS name of the machine hosting your WebPass and Access Manager, *domain* is the name of the server domain to which the machine belongs, and *port* is the number of the port to which Access Manager listens.

2. Navigate to the Authorization Definitions page of the policy domain you want to change:

Access Manager > My Policy Domains > *PolicyName* > Authorization Rules

where *PolicyName* refers to the policy domain you created specifically for impersonation (*ImpersonationPolicyDomain* in our example).

Currently defined authorization rules are listed. If none are listed, click the Add button and complete the form to create one.

3. Click the link to the rule to which you want to add the impersonation action to expand the description.
4. Click the Actions tab, directly under the Authorization Rules tab.

The Authorization Success page appears, with a separate section for Authorization Success and Authorization Failure. If no actions are identified, you must add them. If actions are provided, you can modify them.

You need to add a header variable named “impersonate” to the Authorization Success Action in the policy domain for impersonation.

5. On the Authorization Success page, click the Add or Modify button.

6. In the Authorization Success area, fill in the Return details.

For example:

**Type:** HeaderVar

**Name:** IMPERSONATE

**Return value:** *uid* or *samaccountname* (Active Directory *username*, the Windows domain user for the desired folder)

where “HeaderVar” is the return Type, “IMPERSONATE” is Name of the header variable for impersonation, and the Return value of *uid* or *samaccountname* is based on the directory being used.

7. Save the rule, which is used for the second WebGate request (for authorization) and may look like the one below.

Oblix • NetPoint Access Manager Logged in user: Rohit Valiveti

SharePointViralImpersonation > Authorization Rules > AllowAll > Actions

General Resources **Authorization Rules** Default Rules Policies Delegated Access Admins

General Timing Conditions **Actions** Allow Access Deny Access

**Actions** Custom Actions

**Authorization Success**

Return	Type	Name	Return Attribute
	HeaderVar	IMPERSONATE	uid

Update Cache

Modify Delete

## Adding an Impersonation dll to IIS

You are ready to configure IIS by adding the IISImpersonationExtension.dll to your IIS configuration.

### To add the impersonation dll to your IIS configuration

1. Select Start > Administrative Tools > Internet Information Services (IIS) Manager.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.
3. Click Web Service Extensions on the tree in the left pane.
4. Double-click Oblix WebGate in the right panel to open the Properties panel.
5. Click the Required Files tab.
6. Click Add.

- In the Path to file text box, type the full path to IISImpersonationExtension.dll.  
By default, the path is:  
`WebGate_install_dir\access\oblix\apps\webgate\bin\IISImpersonation\Extension.dll`  
where `WebGate_install_dir` is the directory of your WebGate installation.

---

**Note:** If any spaces exist in the path (for example, C:\Program Files\NetPoint\...) surround the entire string with double quotes (“ ”).

---

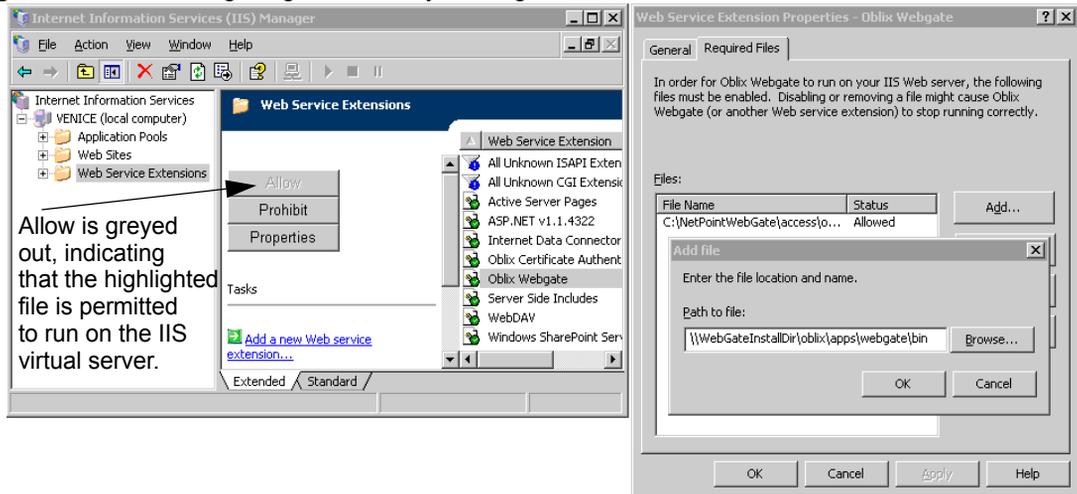
- Click OK.
- Click the General tab on the Web Services Extension Properties panel, then verify that the box “Do not check the file location” is not checked.
- Verify that the Allow button to the left of the Oblix WebGate icon is greyed out, as shown below, which indicates that the dll is allowed to run as a Web service extension.

---

**Note:** If Allow is not greyed out, click it so that it becomes greyed out.

---

**Figure 21** Configuring IIS Security Settings



## Testing Impersonation

You can test Impersonation in the following two ways:

- Testing impersonation outside the SPPS context or test SSO, as described in Table on page 602
- “Testing Impersonation Using the Event Viewer” on page 602
- “Testing Impersonation using a Web Page” on page 603

## **Creating an IIS Virtual Site Not Protected by SPPS**

To test the impersonation feature outside the SPPS context or to test SSO, you will need a target Web page on an IIS virtual Web site that is not protected by SPPS. You create such a virtual Web site by completing the following task.

### **To create an IIS virtual site not protected by SPPS**

1. Select Start > Administrative Tools > Internet Information Services (IIS) Manager.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.
3. Right-click Web Sites on the tree in the left pane, then navigate to New > Web Site on the menu.
4. Respond to the prompts by the Web site creation wizard.
5. After you create the virtual site, you must protect it with a NetPoint Policy domain, as described in the *NetPoint Administration Guide*.

## **Testing Impersonation Using the Event Viewer**

When you complete impersonation testing using the Windows 2003 Event Viewer, you must configure the event viewer before conducting the actual test.

### **To test impersonation through the Event Viewer**

1. Select Start Menu > Event Viewer.
2. In the left pane, right-click Security, then click Properties.
3. Click the Filter tab on the Security property sheet.
4. Verify that all Event Types are checked, and the Event Source and Category lists are set to All, then click OK to dismiss the property sheet.

Your Event Viewer is now configured to display information about the HeaderVar associated with a resource request.



2. Create an IIS virtual site, or use the one you created for the previous task.
3. Place a .asp page or perl script (such as the sample in the preceding listing) anywhere in the tree of the new virtual site.
4. Point your browser at the page, which should appear, with both AUTH\_USER and IMPERSONATE set to the name of the user making the request.

## Completing NetPoint to SPPS Integration

You need to complete several procedures to set up a NetPoint/SPPS integration.

### **Task overview: Setting up NetPoint/SPPS Integration includes**

1. “Configuring IIS Security” on page 604.
2. “Configuring the Wildcard Extension” on page 605 for each SPPS virtual server for which you wish to enable integration.
3. “Editing web.config” on page 607.
- 4.
5. “Testing Your Integration” on page 608

## Configuring IIS Security

Be sure to configure IIS Security before you continue.

### **To configure IIS Security for the SPPS integration**

1. Select Start > Administrative Tools > Internet Information Services (IIS) Manager.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.
3. Click Web Sites on the tree in the left pane.
4. Right-click the icon on the tree in the left pane that represents the SPPS server you are protecting with your WebGate, then select Properties from the drop-down menu.
5. In the property sheet for the SPPS server, click the Directory Security tab.
6. In the Authentication and access control section of the Directory Security tab, click Edit.

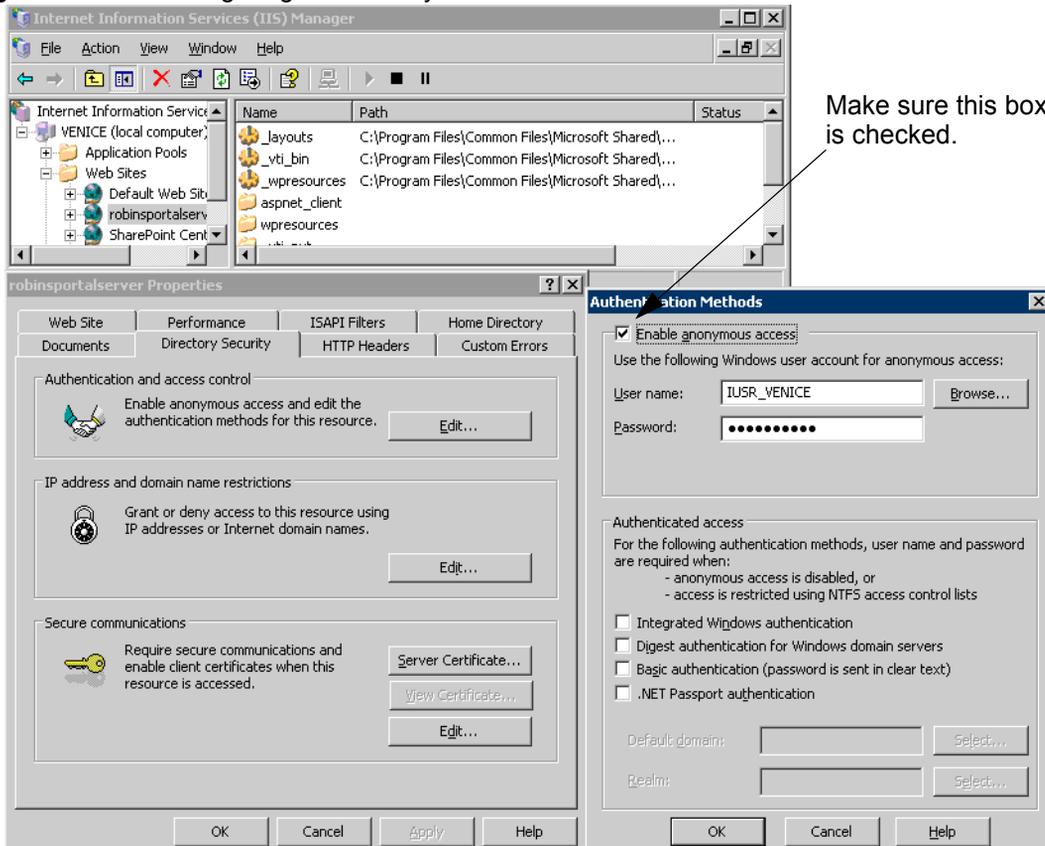
7. In the Authentication Methods panel, click the box labelled Enable anonymous access so that a check appears, then click OK to complete the task.

---

**Note:** Enable anonymous access does not enable anonymous users to access the SharePoint Portal. Rather, this setting configures IIS to relinquish access control to the NetPoint Access System.

---

**Figure 23** Configuring IIS Security



## Configuring the Wildcard Extension

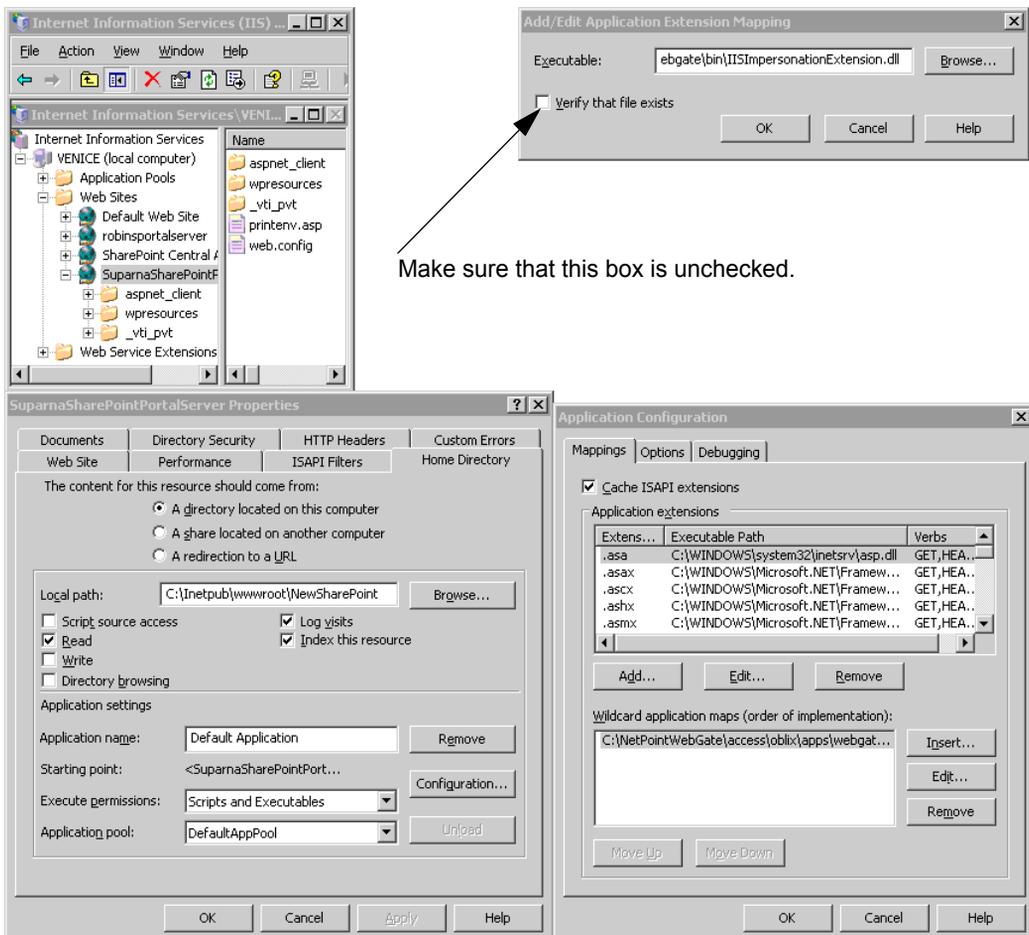
You are ready to configure the wildcard extension for each SPPS virtual server for which you wish to enable integration.

### To configure the wildcard extension for SPPS virtual servers

1. Select Start > Administrative Tools > Internet Information Services (IIS) Manager.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.

3. Click Web Sites on the tree in the left pane.
4. Right-click the icon representing your SPPS server, then click Properties on the drop-down menu.
5. Click the Home Directory tab.
6. Click the Configuration button.
7. In the list box for Wildcard application maps, click the entry for IISImpersonationExtension.dll to highlight it, then click Edit.
8. Ensure that the box is unchecked.
9. Verify that the file exists, then click OK three times to close the Add/Edit panel, the Application Configuration panel and the property sheet for your portal server.

**Figure 24** Configuring the Wildcard Extension



## Editing web.config

You need to add the following line to the web.config file.

```
<add key = "SPS-EnforceIIAnonymousSetting" value="false" />
```

### To edit web.config for the SPPS integration

1. Open Windows Explorer and navigate to the document root of your IIS Website.
2. Use any text editor to open the XML file web.config.
3. Locate the appSettings markers at the end of the file, or create them if they do not exist:

```
<Configuration>  
// [Various configuration settings]  
<appSettings>  
// [Insert "<add key . . .>" here.]  
</appSettings >  
</Configuration>
```

---

**Important:** The appSettings markers are case sensitive and must appear as appSettings.

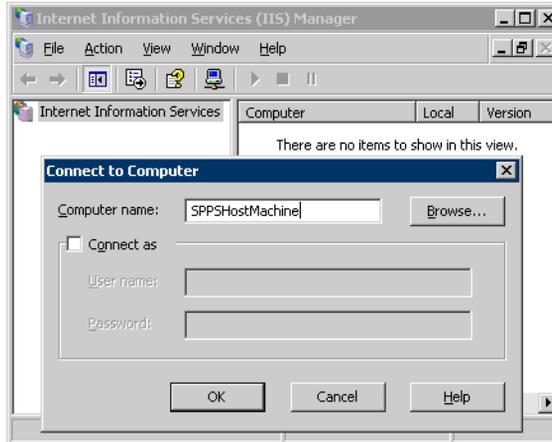
---

4. Add the following line where indicated in the previous listing:  

```
<add key = "SPS-EnforceIISAnonymousSetting" value="false" />
```
5. Save web.config.
6. Restart IIS so that the new setting will take effect by completing the following steps:
  - a) Select Start Menu > Internet Information Services (IIS) Manager.
  - b) In the tree in the left pane, locate the name of the local computer hosting your SPPS installation and write it down; you will need the name of this computer to restart IIS.
  - c) Right-click the local computer icon and select Disconnect from the drop-down menu.
  - d) After the warning asks if you really want to disconnect, click Yes to confirm the action.  
  
The local computer icon disappears from the tree in the left pane, indicating that IIS has been shut down on that machine.
  - e) In the tree in the left pane, right-click the Internet Information Services icon and click Connect on the drop-down menu.

- f) In the Connect to computer panel, type the name of the computer hosting your SPPS installation, then click OK to restart IIS.

**Figure 25** Restarting IIS after Editing web.config



## Uploading User Data

If your COREid installation is configured for any directory server other than SharePoint Active Directory, you must load the user profiles on the other directory server to the SharePoint Active Directory. Do this after you have edited web.config.

## Testing Your Integration

After you complete the tasks to enable integration, you should test to verify that integration is working.

- “Testing NetPoint/SPPS Integration” on page 608
- “Testing SSO for the NetPoint/SPPS Integration” on page 609

## Testing NetPoint/SPPS Integration

You want to verify that a user can access SPPS resources through NetPoint authentication and SPPS authorization.

### To test your NetPoint/SPPS integration

1. Navigate to any SPPS Web page using your browser.  
NetPoint challenges you for credentials.
2. Log in by supplying the necessary credentials, then verify that the page you requested is visible.

3. **Optional**—Check the Event Viewer to confirm that the access request was successful.

## **Testing SSO for the NetPoint/SPPS Integration**

You should also test SSO by demonstrating that a user who has just supplied credentials and accessed an SPPS resource can (before the ObSSO cookie expires) access a non-SPPS resource without having to supply credentials a second time. For example, use a NetPoint resource.

When SSO is working, you should be granted access to the page without having to supply credentials a second time.

### **To test SSO for your NetPoint/SPPS integration**

1. Create and protect a new virtual site with a NetPoint policy domain (or use one you have already created).
2. Place a Web page anywhere in the tree of this virtual site.
3. Using a browser, navigate to the page in the new virtual site.

If you have already passed authentication, you should be granted access to the page without having to supply credentials a second time.



# 16 Integrating NetPoint and the Content Management Server

This chapter explains how to integrate NetPoint 7.0 with the Microsoft® Content Management Server (MCMS) 2002 and covers the following topics:

- “About NetPoint and the MCMS” on page 611
- “Supported Platforms and Requirements” on page 612
- “Request Processing by the Integration” on page 613
- “Integrating NetPoint with the MCMS” on page 614

## About NetPoint and the MCMS

NetPoint provides a full range of identity management and security functions, including: Web-based single sign-on (SSO), user self-service and self-registration, user provisioning, reporting and auditing, policy management, dynamic groups, and delegated administration. NetPoint integrates with all leading directory servers, application servers, Web servers, and enterprise applications.

The Microsoft Content Management Server (MCMS) is an enterprise Web content management system for authoring and delivery. The MCMS streamlines the Web publishing process, enables you to build, deploy, and maintain content-rich Web sites, and allows users to manage their own content. The role-based distributed publishing model of the MCMS includes a multi-level approval workflow, automatic content scheduling and archiving, and content indexing. Developers can create Content Management Server–based applications using ASP.NET and the Microsoft .NET Framework.

The MCMS provides its own authentication mechanisms that leverage IIS and may require an additional login. After integrating NetPoint with MCMS, NetPoint handles authentication and single-sign on (SSO) with the site created using MCMS. NetPoint authenticated users enjoy SSO access to MCMS resources and to NetPoint-protected resources.

The integration of MCMS with NetPoint requires authentication schemes based on Windows Impersonation. In addition, NetPoint supports URL-level authorization. MCMS performs application-level authorization based on the roles you set up in the MCMS.

The MCMS is often used with the Microsoft SharePoint Portal Server (SPPS) for developing and managing Web content. The Microsoft Content Management Server 2002 Connector for SharePoint Technologies enables you integrate the Content Management Server with the Microsoft Office® SharePoint Portal Server. The connector enables sharing of key publishing and search technologies. For details about integrating NetPoint with the SharePoint Portal Server, see “Integrating NetPoint SSO with SharePoint Portal Server” on page 585.

## About Windows Impersonation

NetPoint to MCMS integration relies on the Windows impersonation feature, which enables a trusted user in the Windows server domain to assume the identity of any user requesting an MCMS target resource. This trusted impersonator maintains the identity context of the user while accessing the resource on behalf of the user.

Impersonation is transparent to the user; access appears to take place directly, as if the MCMS resource were a resource within the NetPoint domain. For more information, see “Setting Up Impersonation” on page 595.

## Supported Platforms and Requirements

Successful NetPoint integration with MCMS requires both NetPoint and Microsoft components, which must be installed and configured to support impersonation as well as integration. See the following topics for requirements:

- “Required NetPoint Components” on page 612
- “Required Microsoft Components” on page 613

## Required NetPoint Components

The following NetPoint 7.0 components are required to integrate NetPoint with MCMS. With the exception of a WebGate, all NetPoint components may reside on different machines or the same machine as the MCMS.

- COREid Server
- WebPass
- Access Manager
- Access Server

- WebGate installed with the MCMS on a Windows Server 2003

The NetPoint 7.0 ISAPI WebGate includes the IISImpersonationExtension.dll, which you need to configure manually to enable impersonation for the MCMS integration.

The NetPoint 7.0 IISImpersonationExtension.dll is an IIS wildcard extension that checks whether the Authorization Success Action headerVar has been set to impersonate. If it has been, the dll creates a Kerberos U4S2Self ticket so that the special trusted user in the MCMS Active Directory can impersonate the user who originally made the request.

## Required Microsoft Components

NetPoint 7.0 supports the Microsoft Content Management Server 2002 with Service Pack 1a on the following platform:

- Windows Server 2003 Enterprise Edition
- Microsoft IIS v6.0 Web Server
- Active Directory (the domain controller must be on a Windows 2003 Server)
- MSSQL supported by the MCMS
- **Optional**—Microsoft SharePoint Portal Services

## Request Processing by the Integration

NetPoint uses the Windows impersonation feature to facilitate user access to MCMS resources.

### **Process overview: Request processing with MCMS integration**

1. The user requests access to an MCMS resource.
2. The WebGate protecting MCMS intercepts the request, determines whether the target resource is protected, and if it is, challenges the user for authentication credentials.
3. The user supplies credentials and the Access Server validates them.

4. Upon validation, the WebGate sets an ObSSO cookie in the user's browser, thus enabling SSO.

The WebGate also sets an HTTP header variable called impersonate, whose value is set to the authenticated user's LDAP uid (*samaccountname*, if the user account exists in Active Directory, or *userPrincipalName*, if the user account exists in a multi-domain Active Directory forest).

---

**Note:** At this point, IIS considers the user to be anonymous, since the impersonation has not yet been set.

---

5. The NetPoint ISAPI wildcard extension IISImpersonationExtension.dll checks for the Authorization Success Action header variable named impersonate.
6. When such a header variable exists, the wildcard extension obtains a Kerberos ticket for the user.

This Service for User to Self (S4U2Self) impersonation token enables the designated trusted user to assume the identity of the requesting user and obtain access to the target resource through IIS and MCMS.

7. Authorization is performed by the MCMS based on the roles setup in the MCMS.
8. When authorization is successful, the user is granted access to the resource.

## Integrating NetPoint with the MCMS

You need to complete several procedures to integrate NetPoint with the Content Management Server.

---

**Note:** The procedures in this chapter illustrate how to integrate NetPoint with the MCMS using a sample Web site (the Microsoft WoodgroveASP Web site).

---

### Task overview: Integrating NetPoint with MCMS

1. "Installing NetPoint" on page 615
2. "Installing Microsoft Components" on page 615
3. "Integrating NetPoint with the MCMS" on page 616
4. "Setting Up Impersonation" on page 616
5. "Completing the NetPoint to MCMS Integration" on page 617
6. "Testing the NetPoint to MCMS Integration" on page 618

## Installing NetPoint

The ISAPI Webgate for MCMS must be installed on the machine that hosts the MCMS. All other NetPoint components can reside together on the machine hosting the MCMS or on any other machine.

If both NetPoint and MCMS are set up for different instances of Active Directory, both instances must belong to the same Active Directory domain.

### To install NetPoint for the integration

1. Install a COREid Server and a WebPass, then set up the COREid System, as described in the *NetPoint 7.0 Installation Guide*.
2. Install and set up the Access Manager and one or more instances of the Access Server, as described in the *NetPoint 7.0 Installation Guide*.
3. Install WebGates, as described in the *NetPoint 7.0 Installation Guide*.

---

**Note:** One WebGate must be installed on the machine hosting the MCMS, as described in “Integrating NetPoint with the MCMS” on page 616.

---

## Installing Microsoft Components

Except where noted, all MCMS components from Microsoft must be installed on the same host machine.

### To install Microsoft components

1. On a machine Windows Server 2003 with IIS v6.0, complete activities below to install the MCMS using instructions in your Microsoft documentation:
  - a) Create Windows user accounts.
  - b) Create a database in MSSQL and grant rights to the system administrator account.
  - c) Create two Web sites.
  - d) Install the MCMS 2002 SP1a.
  - e) Configure the database with the MCMS 2002 Database Configuration Application (DCA).
  - f) Configure the MCMS server using the Server Configuration Application (SCA).
  - g) Update the maximum upload size settings in the web.config file.
  - h) Install Site Manager.
2. On a Windows 2003 Server host, install Active Directory for the MCMS using instructions in your Active Directory documentation.

3. Ensure your MCMS installation is working properly using instructions in your Microsoft documentation as you:
  - a) Download a sample WoodGroveASP Web site and install it on the MCMS site to use as a test vehicle for the procedures in this chapter.
  - b) Ensure you can log into the Site Manager, Server Configuration Application, and the sample WoodgroveASP Web site.
4. **Optional**—Install the Microsoft Content Management Server 2002 Connector for SharePoint Technologies, as described in your Microsoft documentation.

For details about integrating NetPoint and the SPPS, see “Integrating NetPoint SSO with SharePoint Portal Server” on page 585.

## Integrating NetPoint with the MCMS

After installing NetPoint and the MCMS, as described earlier, you need to complete the steps below to integrate the two environments.

### To integrate NetPoint with the MCMS

1. On the Windows 2003 Server machine hosting the MCMS, install a NetPoint ISAPI WebGate using instructions in the *NetPoint 7.0 Installation Guide*.  
The IISImpersonationExtension.dll is installed automatically in:  
`webGate_install_dir\access\oblix\apps\webgate\bin\`  
where *WebGate\_install\_dir* is the directory where you installed the WebGate.
2. Install a NetPoint 7.0 WebGate on the MCMS site, and impersonation.dll in the WoodGroveASP site.
3. See “Setting Up Impersonation” on page 616.

## Setting Up Impersonation

The integration between NetPoint and the MCMS requires Windows impersonation.

---

**Note:** The procedures to set up impersonation are outlined below. You should be able to use the procedures provided in the chapter on impersonation in the *NetPoint 7.0 Administration Guide Volume 2* and in “Integrating NetPoint SSO with SharePoint Portal Server” on page 585 to implement impersonation in your environment. Details are not repeated in this chapter.

---

## **Task overview: Setting up impersonation for the MCMS**

1. Create a trusted user account for only impersonation in the Active Directory connected to MCMS, as described in “Creating a Trusted User Account” on page 596.
2. Give the trusted user the special right to act as part of the operating system, as described in “Assigning Rights to the Trusted User” on page 597.
3. Bind the trusted user to the WebGate by supplying the authentication credentials for the trusted user, as described in “Binding the Trusted User to Your WebGate” on page 598.
4. Add a header variable named impersonate to Authorization Success Action in the policy domain for impersonation, as described in “Adding an Impersonation Action to a Policy Domain” on page 599.
5. Configure IIS by adding IISImpersonationExtension.dll to your IIS configuration, as described in “Adding an Impersonation dll to IIS” on page 600.
6. Test impersonation, as described in “Testing Impersonation” on page 601.
7. Proceed as described in “Completing the NetPoint to MCMS Integration” on page 617.

## **Completing the NetPoint to MCMS Integration**

After confirming that impersonation is set up properly, you need to perform steps below to complete the integration, ensure that everything is working properly, and confirm that you have single-sign on access.

### **To complete the NetPoint to MCMS integration**

1. Move the two ISAPI filters installed at the IIS Top Level Website by MCMS to the two virtual Web sites created for MCMS, as indicated below:

#### **MCMS HTML Packager Filter**

C:\Program Files\Microsoft Content Management Server\Server\bin\REHTMLPackager.dll

#### **MCMS ISAPI Filter**

C:\Program Files\Microsoft Content Management Server\Server\bin\REAuthFilt.dll

2. Complete steps below to finish the impersonation implementation for the MCMS integration:
  - a) “Configuring IIS Security” on page 604 for this environment the impersonation implementation.

- b) “Configuring the Wildcard Extension” on page 605 for each MCMS virtual server for which you wish to enable integration.
3. Give appropriate rights to users for viewing different sections of the Web site using the Site Manager.
4. Using the NetPoint Access Manager, create Policies to protect the WoodGroveASP top-level resource.

## Testing the NetPoint to MCMS Integration

After you complete the tasks to enable integration, it is a good idea to test the integration to verify things are working as expected.

- “Testing NetPoint/MCMS Integration” on page 618
- “Testing SSO for the NetPoint/MCMS Integration” on page 619

## Testing NetPoint/MCMS Integration

It is important to verify that a user can access MCMS resources through NetPoint authentication and MCMS authorization.

### To test your NetPoint/SPPS integration

1. Navigate to a WoodGroveASP Web site using your browser.  
NetPoint challenges you for credentials.
2. Log in by supplying the necessary credentials.
3. Confirm that you have access.
4. **Optional**—Check the Event Viewer to confirm that the access request was successful.

## Testing SSO for the NetPoint/MCMS Integration

You test SSO by demonstrating that a user who has just supplied credentials and accessed an MCMS resource can (before the ObSSO cookie expires) access a non-MCMS resource without having to supply credentials a second time. For this test you can use a NetPoint resource.

When SSO is working, you should be granted access to the page without having to supply credentials a second time.

### To test SSO for your NetPoint/MCMS integration

1. Create a new resource and protect it with a NetPoint policy domain (or use one you have already created).
2. Using a browser, navigate to the resource.

If you have already passed authentication, you should be granted access to the page without having to supply credentials a second time.



# 17 Integrating NetPoint with OctetString VDE

This chapter focuses on integrating NetPoint with the OctetString Virtual Directory Engine (VDE) to enable COREid Data Anywhere. It includes the following topics:

- “About NetPoint and VDE Integration” on page 622
- “Integration Limitations” on page 630
- “Integration Architecture” on page 632
- “About Schema Extension” on page 635
- “Integration Scenarios and Limitations” on page 639
- “Integration Requirements” on page 644
- “About the NetPoint-VDE Integration Process” on page 649
- “Preparing Your Environment” on page 652
- “Installing and Configuring VDE and DME” on page 657
- “Installing the First NetPoint COREid Server” on page 663
- “Extending Directory Schemas” on page 665
- “Creating Mapping Files for Adapters” on page 667
- “Creating Data Store Adapters” on page 669
- “Customizing Adapters and Mapping Files” on page 679
- “Completing NetPoint Installation and Setup” on page 698
- “Testing Your Integration” on page 699
- “Reference Information” on page 700
- “NetPoint-VDE Integration Templates” on page 708
- “Integration Tips” on page 717
- “Troubleshooting” on page 721

The NetPoint-VDE integration uses only a subset of VDE functions; therefore not all of the information provided in the OctetString documentation applies to the NetPoint-specific configurations described in this chapter.

This chapter should be used in conjunction with the manuals below:

- *VDE 3.0 Installation Guide*
- *VDE 3.0 Product Manual*

## About NetPoint and VDE Integration

The OctetString Virtual Directory Engine (VDE) combines the user data from multiple data sources to create an aggregated, *virtual directory*.

From the point of view of NetPoint applications, the virtual directory looks and behaves just like any other LDAP directory, and the NetPoint user usually does *not* receive any obvious indications that the data retrieved by NetPoint has come from heterogeneous sources.

From the perspective of the target data store owners, the impact of VDE is minimal; the data store owners do *not* relinquish ownership of their data, VDE does *not* reformat the native data structures, and *no* permanent copies of the original data are maintained by VDE.

To enable certain NetPoint features, you must extend the schema of the target LDAP directories or add columns simulating NetPoint auxiliary user attributes to the primary database tables included in your virtual directory. For more information, see “About Schema Extension” on page 635.

## Key Terms and Features

To explain precisely the issues surrounding NetPoint-VDE integration, this document uses the following terms in very specific fashion.

### Terms

**Virtual Directory**—A logical, aggregated directory that presents user data drawn from multiple sources, just as if all that data came from a standard LDAP directory to which a customer-defined schema has been uniformly applied. For the purposes of NetPoint integration, VDE does not create permanent copies of user profiles outside the native data sources. Rather, VDE retrieves and transforms each user profile as it is requested by a NetPoint application.

You can configure your virtual directory as a single, contiguous searchbase or as multiple disjoint searchbases. For details, see “About Searchbase Options” on page 626.

**Super Directory**—A special type of *virtual directory* that facilitates namespace mapping. It can contain any combination of federated LDAP directories, RDBMS databases, and embedded virtual data sources. The embedded virtual data sources

can be *split profiles*, native RDBMS Joins, and native RDBMS Views. The super directory, which is the only supported method for producing a single, contiguous searchbase aggregated from multiple data stores, connects to NetPoint by means of a VDE local store adapter.

**Federation**—A method by which VDE makes a data source visible in the virtual directory it presents to NetPoint. All the data for a given user profile comes from a single data store such as an LDAP directory, a single-table database, or an embedded virtual data source.

Different user profiles can come from different *federated data stores*, which incorporate any combination of the following types of data sources:

- Multiple, heterogeneous LDAP directories
- Multiple relational databases that store all user data in a single table
- Embedded virtual data sources, which fall into the following three categories:
  - Split profiles involving any combination of directories and databases
  - Native RDBMS Views involving multiple database tables
  - Native RDBMS Joins involving multiple database tables

For more information, see “Federated Data Stores” on page 625.

**Embedded Virtual Data Source**—A virtual object that VDE “sees” as a target data store it can present to NetPoint or federate in a virtual directory, then present to NetPoint. Each embedded virtual data store aggregates two or more target data stores. The three types of embedded virtual data stores are:

- Split profile
- Native RDBMS Join
- Native RDBMS View

In general, embedded virtual data stores are suitable for authentication and authorization activities only, because they necessarily involve secondary data sources, which are sometimes not available for the full range of NetPoint identity management activities.

**Split Profile**—A special type of embedded virtual data source created from more than one data source. Split Profiles draw the user profile attributes for each user from multiple sources, including LDAP directories and multiple database tables.

Each data store contributes some of the attributes necessary to complete the full set of user profile attributes that gets mapped into the VDE virtual directory. These attributes can come from LDAP directories or database tables. All NetPoint user attributes must reside in the primary data store, because not all NetPoint operations can be performed on the attributes in the secondary stores. VDE can make a split

profile visible to NetPoint as a standard LDAP directory. Alternatively, a split profile can be federated as part of a virtual directory.

For more information, see “Split Profiles” on page 627 and Figure 33 on page 632.

**Single-Table Database**—A single-table database does not necessarily refer to a database that contains just one table, but rather, a database that stores in just one table all the user profile attributes that get mapped into the top level virtual directory.

**Multi-Table Database**—A database that stores in more than one table the user profile attributes that get mapped into the virtual directory.

**Virtual Directory Schema**—This is the schema developed by the customer for use by the top-level directory that VDE makes visible to NetPoint. It must be extended with the NetPoint attributes. See “Virtual Directory Schema” on page 637.

Optionally, you can further extend the virtual directory schema with customer attributes drawn from the target data sources. For details, see “Customer Schemas” on page 638.

## Features

Virtual directory technology enhances NetPoint capabilities with four major features:

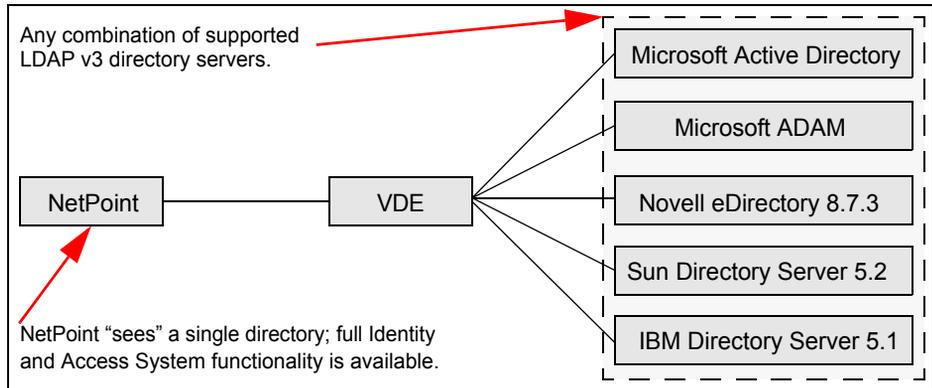
- **Federated Data stores**, mentioned above and discussed in more detail in “Federated Data Stores” on page 625
- **Split Profiles**, mentioned above and discussed in more detail in “Split Profiles” on page 627
- **Aggregated Namespaces**—Map target data stores and embedded virtual data sources to nodes in the super directory. You must install a local store adapter to create the super directory. For details, see “Aggregated Namespaces” on page 628.
- **Schema Mapping**—Transforms the data from all the target data stores according to the customer-defined schema in the top-level virtual directory. For details, see “Aggregated Schema Mapping” on page 629.

For background discussion of the general advantages offered by virtual directories, see the *VDE 3.0 Product Manual*.

## Federated Data Stores

The VDE virtual directory allows NetPoint users to access and manipulate user data from disparate, multiple sources, just as if all user accounts came from a single, uniformly schematized data store. It can incorporate user data from LDAP directories even if the host directory servers come from different vendors and use different schema. Figure 26 illustrates how NetPoint connects to multiple LDAP directories.

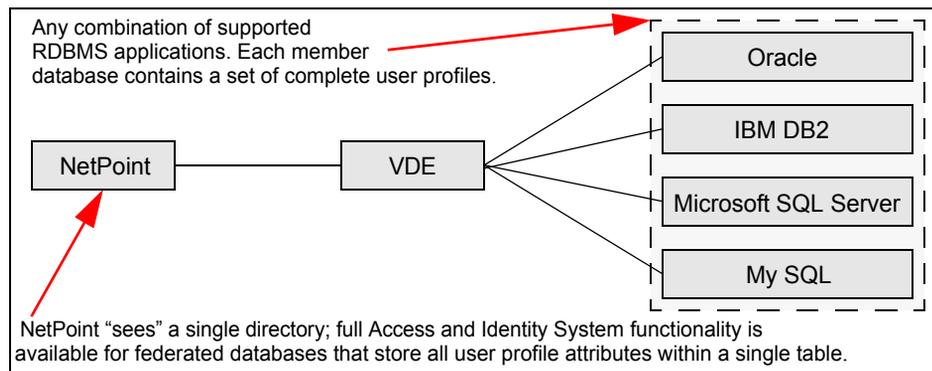
**Figure 26** VDE Integration Involving Federated LDAP Directories



Your VDE virtual directory can also incorporate RDBMS databases that store all user data in a single table. For details on integrating databases that spread user data across multiple tables, see “Split Profiles” on page 627.

Figure 27 illustrates how NetPoint connects to multiple relational databases. See also “Database Connectivity Tips” on page 719.

**Figure 27** VDE Integration Involving Federated RDBMS Applications



## About Searchbase Options

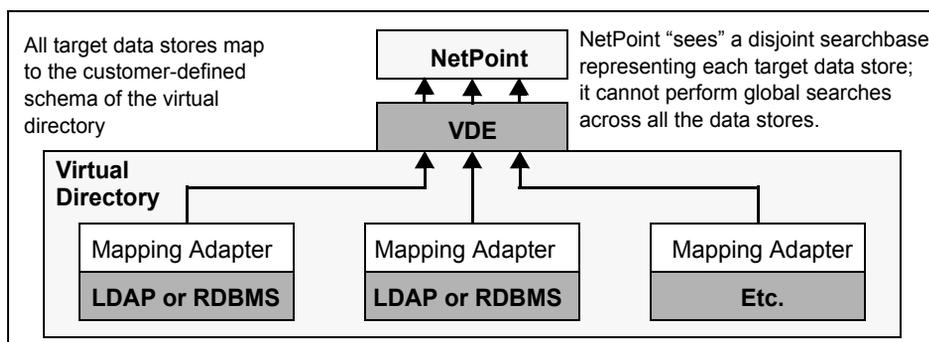
NetPoint supports two options for federating target data stores through VDE:

- Disjoint Searchbases
- Unified Searchbase

**Disjoint Searchbases**—You can configure your NetPoint-VDE integration so that NetPoint “sees” each target data store as a distinct, disjoint searchbase within the virtual directory. Namespace aggregation is not possible for such a configuration. Also, each target data store resides behind a different top-level mapping adapter, so global directory searches across all the data sources are not possible.

Figure 28 illustrates a virtual directory configured for disjoint searchbases.

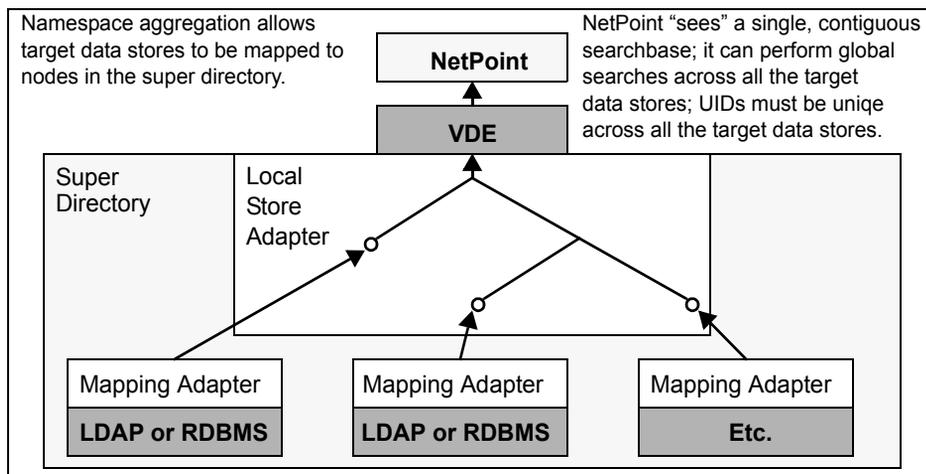
**Figure 28** A Virtual Directory with Disjoint Searchbases



**Unified Searchbase**—You can create a super directory by installing a Local Store Adapter at the top level, then creating nodes to which you map your target data stores. This option allows for both global directory searches and powerful namespace aggregation.

Figure 29 provides an illustration of super directory with a unified, contiguous searchbase.

**Figure 29** A Super Directory with a Unified, Contiguous Searchbase



## Split Profiles

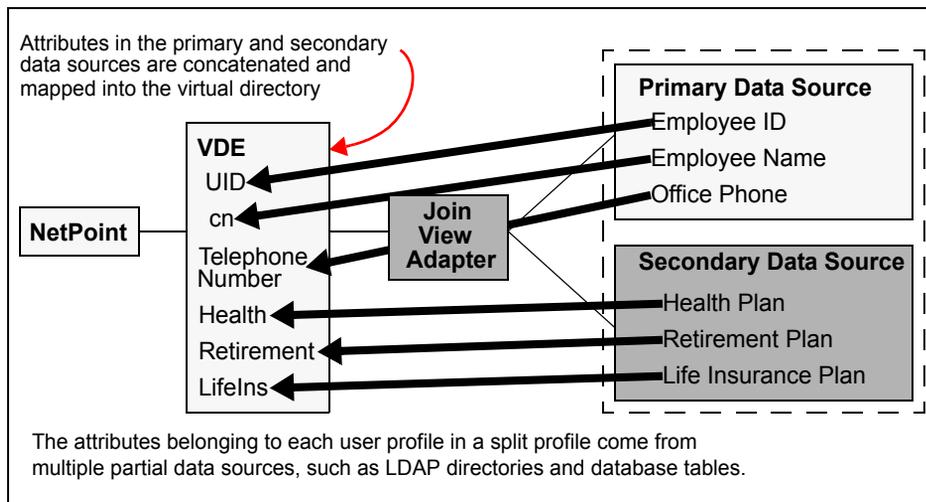
In addition to providing NetPoint users with access to federated data stores, VDE can provide access to virtualized *split profile* data sources which draw user profile attributes from multiple data sources, such as LDAP directories and relational database tables.

For example, you can store attributes such as user login password and office phone number in an Active Directory account maintained by Information Technology, while storing other attributes such as home phone number and health plan affiliation in a relational database account maintained by Human Resources.

Because this distribution of attributes across multiple data sources precludes the execution of certain NetPoint identity management functions on secondary data stores, split profile configurations are suitable primarily for authentication and authorization (Access System) operations.

Figure 30 illustrates a simple NetPoint-VDE integration involving a split profile.

**Figure 30** NetPoint-VDE Integration for a Simple Split Profile



The primary data source contains the NetPoint user branch schema attributes, while the secondary data sources usually contain customer attributes.

All Access System and Identity System operations can be performed on the attributes in the primary data source. All Access System operations can also be performed on the data in the secondary sources, but certain Identity System operations can not be performed on attributes residing in the secondary data stores.

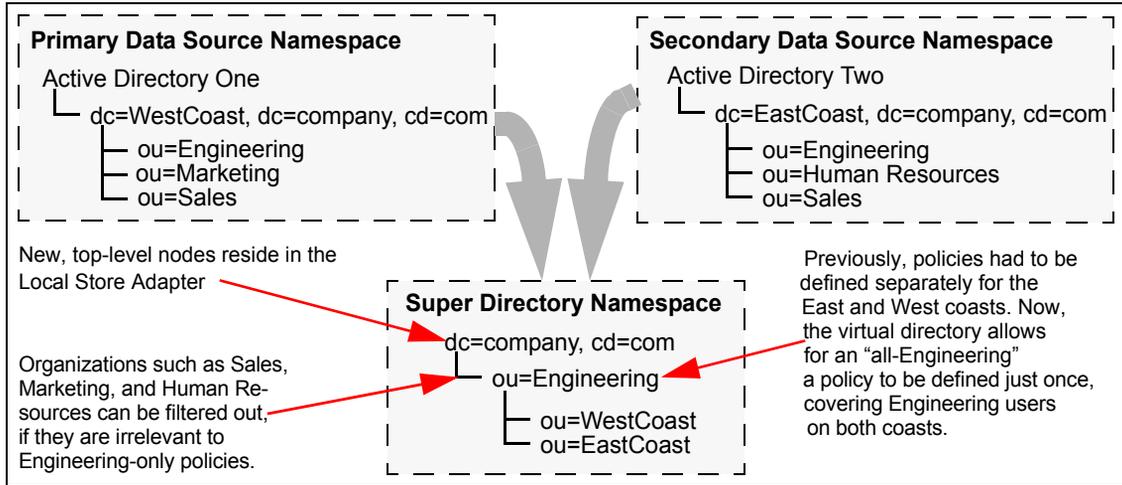
For more information, see “Integration Limitations” on page 630.

## Aggregated Namespaces

When you create a super directory, you can specify a namespace hierarchy ideally suited to your NetPoint identity management and policy management needs. This new hierarchy can differ from the native namespace hierarchies used by the constituent data stores in the virtual directory.

As Figure 31 illustrates, attributes can be reorganized and assigned to new levels.

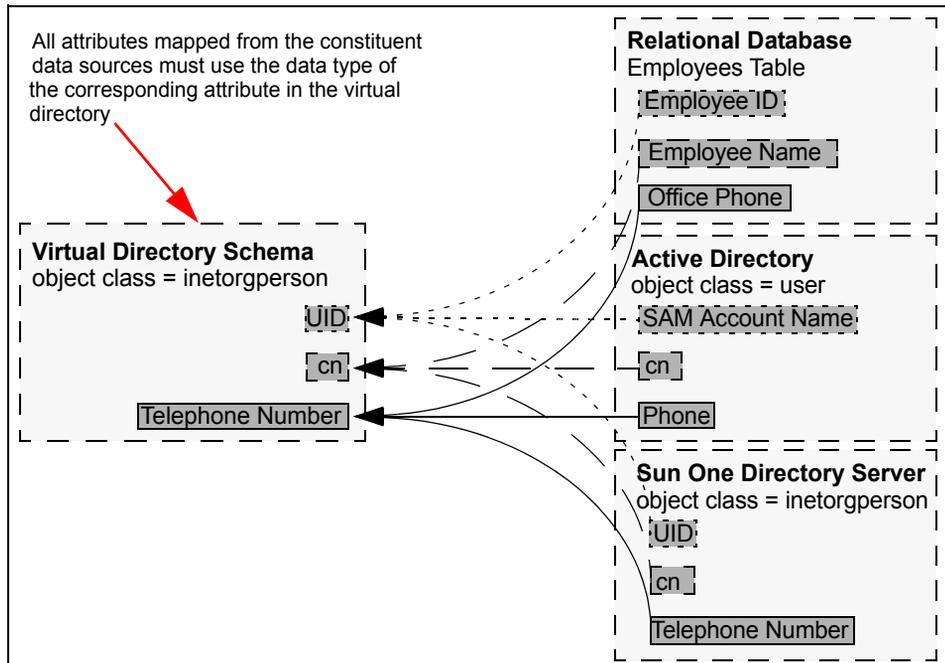
**Figure 31** Namespace Aggregation for a Simple Super Directory



## Aggregated Schema Mapping

When you create your VDE virtual directory, you map the native schema used by the constituent data stores to the schema used by your virtual directory. Figure 32 illustrates this mapping on a simple VDE system.

**Figure 32** Aggregated Schema Mapping for a Simple Virtual Directory



# Integration Limitations

VDE extends NetPoint functionality to multiple, heterogeneous directories and databases, but with certain limitations. When you deploy NetPoint with VDE, you must carefully observe the limitations stated in this document. Table 42 lists the virtual directory configurations subject to limitation and provides references to detailed discussions of these issues.

**Table 42** NetPoint Feature Availability for Virtual Directory Configurations

Data Source	NetPoint Feature Availability	
	Access System	Identity System
Federated LDAP directory	Full	Full
Federated “single table” database	Full	Full
Federated “multi-table” database (using the native RDBMS Join feature)	Full	Full functionality for primary data stores. Add, Modify, and Delete are also available for secondary data stores if the native RDBMS Join feature supports these functions.
Federated “multi-table” database (using the native RDBMS View feature)	Full	Full functionality for primary data stores, but Add and Delete are not available for secondary data stores. (Modify is available for secondary data stores).
Split-Profile directory (using the VDE Join View adapter)	Full	Full functionality for primary data stores, but Add, Modify and Delete are not available for secondary data stores.

For more information, see:

- “About Limitations on Multi-Value Attributes” on page 630
- “About Limitations on Embedded Virtual Data Sources” on page 632

See also “Database Connectivity Tips” on page 719.

## About Limitations on Multi-Value Attributes

Individual attributes stored in standard LDAP directories can take multiple values. For instance, you can record each user’s password history or assign multiple subscriptions to a user account stored in an LDAP directory.

By contrast, properly normalized data tables in SQL-compliant RDBMS applications cannot store multiple values for the same user attribute within a single table. Therefore, NetPoint-VDE integrations involving database tables support

only limited functionality for multi-valued attributes. For details, consult Oblix customer care.

---

**Note:** If your virtual directory incorporates LDAP directories exclusively, no restrictions apply to multi-valued attributes.

---

User profiles already stored in existing RDBMS databases are most likely to have been implemented entirely with single-value attributes, so no restrictions apply, as long as all the database tables you incorporate into your virtual directory contain single-value attributes only.

In rare situations where multi-valued attributes were used to implement the user accounts in a non-normalized RDBMS data store, you can incorporate the unsupported tables containing multi-valued attributes into your virtual directory as long as you carefully observe the following limitations on User and Group Manager operations:

- The password history function is not supported
- No more than one administrator can be configured per group
- No more than one subscription can be configured per group
- Either group subscription or unsubscription notification can be activated, but you can not activate both simultaneously
- No more than one dynamic filter can be configured per group
- No more than one group type can be configured per group
- No more than one subscription type can be configured per group (The possible types are: Open, Close, Open with filter, and Controlled through workflow).
- A database table in the virtual directory can contain no more than one multi-valued attribute.

---

**Important:** If you are creating new data stores for your virtual directory, Oblix strongly recommends that you use LDAP directories whenever possible. This is because the limited ability of relational databases to handle multi-valued attributes restricts the functionality available in your identity management application. If you are working with existing user data stored in a relational database, please familiarize yourself thoroughly with the restrictions on multi-valued attribute handling within the virtual directory.

---

# About Limitations on Embedded Virtual Data Sources

Table 43 lists the limitations on embedded virtual data sources involving multiple database tables.

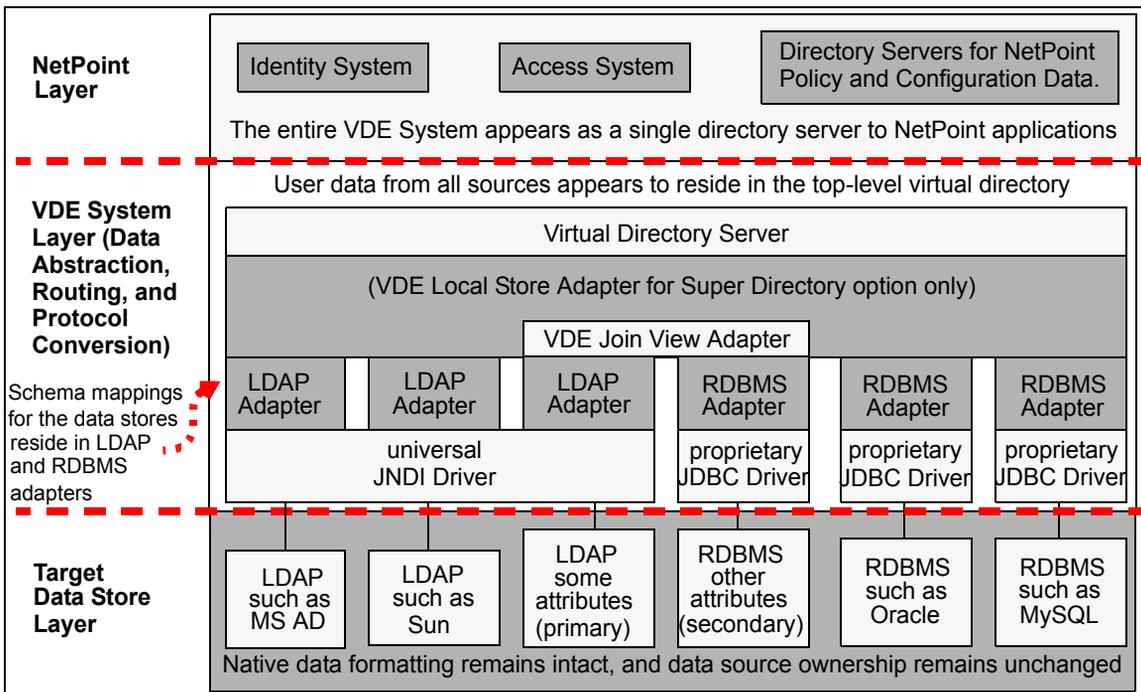
**Table 43** Identity Management Function Availability for Multi-table Configurations

NetPoint Identity Management Function	Table Aggregation Method		
	Join View Adapter (Split Profile)	Native RDBMS Join Feature	Native RDBMS View Feature
Modify	No	Yes, if supported by the native RDBMS Join feature	Yes
Add	No	Yes, if supported by the native RDBMS Join feature	No
Delete	No	Yes, if supported by the native RDBMS Join feature	No

# Integration Architecture

The NetPoint-VDE integration consists of three layers, shown in Figure 33:

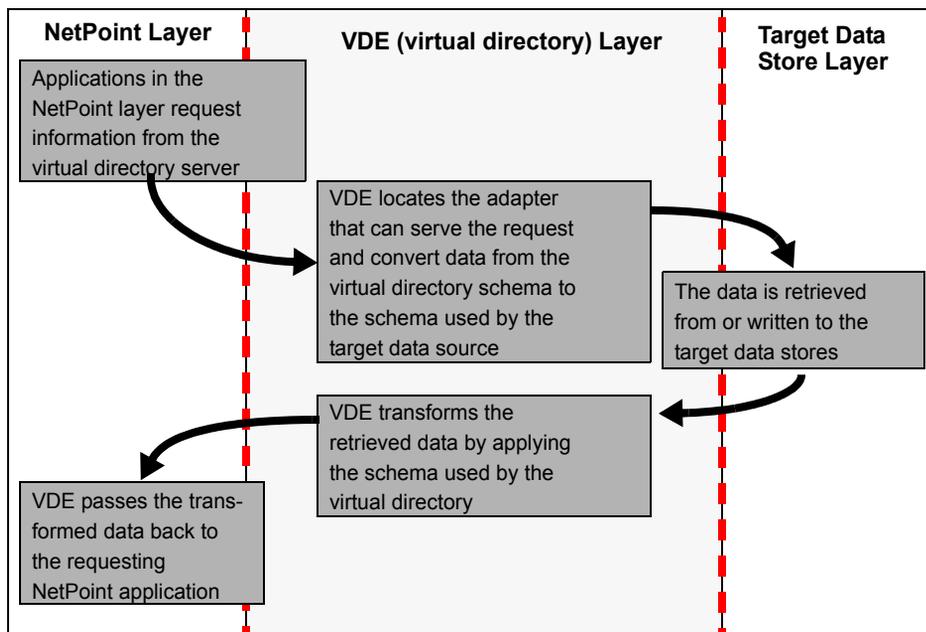
**Figure 33** VDE Integration Layers



To users and applications in the NetPoint layer, the VDE system appears to be a single LDAP directory that includes the standard schema, plus the extended NetPoint attributes.

Within the virtual directory layer, VDE accepts requests for user data from the NetPoint applications, retrieves the requested data from the constituent data stores, transforms that data so that it conforms to the NetPoint schema, then passes the processed data back to the requesting NetPoint application. Figure 34 illustrates the steps in this process.

**Figure 34** Data Request Handling in a Simple NetPoint-VDE Integration



To the administrators of the directories and databases in the target data store layer, the NetPoint-VDE integration appears to have minimal impact, because implementation does not require permanent alteration of the native namespaces or data structures for either directories or databases.

---

**Note:** Depending on the NetPoint features you wish to use, you may need to add certain NetPoint auxiliary attributes as columns in target database tables. You also must extend the target LDAP directory schema with the user branch of the NetPoint (Oblix) schema. For details, see “About Schema Extension” on page 635.

---

Furthermore, the virtual directory does not require that the data be copied permanently to a location beyond the control of the original data owner. Finally, data security is maintained or even enhanced, because access to individual data stores and even individual user profiles can now be controlled through NetPoint Identity System attribute access control.

## About VDE Drivers and Adapters

VDE uses special drivers and adapters to connect to the data sources it incorporates in its virtual directory.

**JNDI Driver**—The JNDI driver is shipped as part of the VDE installation package. A JNDI driver connects VDE to the LDAP directories, and a JDBC driver connects VDE to the RDBMS sources. You install these drivers on the machine that hosts VDE.

**JDBC Driver**—You must install the appropriate version of the JDBC driver for each RDBMS application you use. See the *VDE 3.0 Product Manual* and the *VDE 3.0 Installation Guide* for details.

**Adapters**—In addition to installing the appropriate driver for each data source in your virtual directory, you must configure an LDAP or RDBMS adapter for each directory or relational database that connects to VDE. These adapters contain the mapping information VDE uses to transform user profile information from the native data stores with the virtual directory schema. For details, see “Creating Data Store Adapters” on page 669.

## About NetPoint-Specific Data

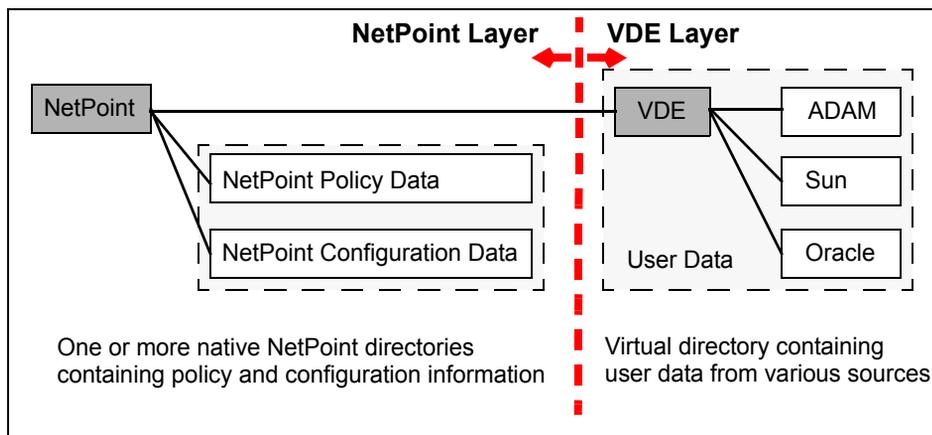
The NetPoint user, policy, and configuration data each occupy an LDAP directory information tree (DIT) branch. The NetPoint-VDE integration requires that each branch exist in a specific location. Table 44 lists these requirements.

**Table 44** Required locations for the branches of the NetPoint directory

Branch	Location
Policy	These branches must reside on one or more directory servers within the NetPoint Layer. The host directories are native to the NetPoint system.
Configuration	
User Data	All user data stores appear to be part of the top-level directory, which resides on the machine hosting VDE within the VDE layer.

Figure 35 illustrates the policy and configuration branch location for a VDE integration.

**Figure 35** Policy and Configuration Branch Location for a VDE Integration



## About Schema Extension

For NetPoint to function properly, you need certain NetPoint attributes like `userid`, `userpassword`, and others to be extended to your schema.

Regardless of the native schemas or table structures used by the data stores in the virtual directory, NetPoint “sees” only the schema used by the VDE virtual directory. This is because VDE automatically maps the native object classes and attributes used by the various data stores to the corresponding logical object classes and attributes used by the virtual directory.

VDE supplies a default virtual directory schema, which is quite similar to the *de facto* industry-standard schema for LDAP directories. You can use this as a starting point when developing a virtual directory schema optimized for the needs of your enterprise.

All files required for schema extension are located in:

```
DN_ConversionTool_install_dir\oblix\tools\DataAnyWhere\OblixUserSchema\
```

---

**Important:** The DNConversion Toolkit is a stand-alone package that you *must* download from the Oblix Web site to ensure that you have the latest version. *Always* use files in the stand-alone DNConversion Toolkit. An *earlier* version of the toolkit is included in the `COREid_702_install_dir`. Do *not* use the toolkit in the `COREid_install_dir`.

---

Figure 36 illustrates both the required and optional schema extension tasks involved with NetPoint-VDE integration.

**Figure 36** Schema Extension Tasks for NetPoint-VDE Integration

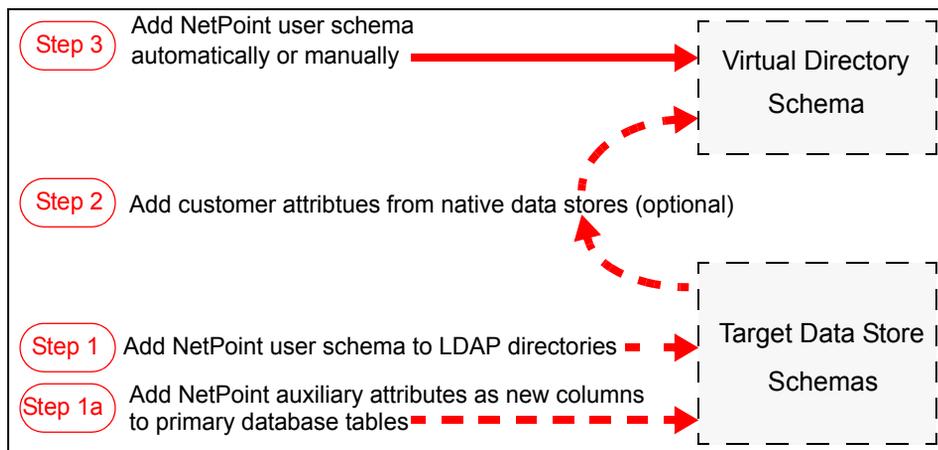


Table 45 lists the schema requirements for various components of the virtual directory.

**Table 45** Schema Requirements for Components of a NetPoint-VDE Integration

Component	Schema Requirements
The virtual directory	Must be extended with the NetPoint user schema. Should also be extended with customer attributes.
LDAP directories connected to VDE as federated data sources	Must be extended with the NetPoint user schema.
Databases connected to VDE as federated data sources	Table columns must be added to the database tables serving as primary data stores. Each column represents a NetPoint auxiliary attribute that enables a NetPoint feature you plan to use. For details on the specific NetPoint features enabled by each attribute in the NetPoint user branch schema, see “NetPoint Auxiliary Attributes” on page 700.
Split profile	The schema of the primary data store must be extended with the NetPoint user schema. If the primary data store is a database table, a column must be added for each NetPoint auxiliary attribute that enables a NetPoint feature you plan to use. For details on the specific NetPoint features enabled by each attribute in the NetPoint user branch schema, see “NetPoint Auxiliary Attributes” on page 700.

For more information, see:

- “Virtual Directory Schema” on page 637
- “Target Directory Schemas” on page 637

- “About Adding Attributes to Target Database Tables” on page 638
- “About Adding Attributes to Target Database Tables” on page 638
- “Customer Schemas” on page 638

## Virtual Directory Schema

To enable your virtual directory to connect to NetPoint and make use of all NetPoint features, you must extend it with NetPoint attributes. For example, NetPoint requires attributes assigned to the Full Name, Login, and Password semantic types for Person and Group object classes. This and other essential NetPoint user data occupies a branch in each NetPoint-enabled user directory. For further discussion, see the section about NetPoint object classes in the chapter on setting up the COREid System in the *NetPoint 7.0 Installation Guide*.

For a detailed listing of the NetPoint schema, see the *NetPoint 7.0 Schema Description*. You can update the VDE schema with NetPoint user attributes in the following two ways:

- **Automatically**—This occurs if you select the “Auto configure objectclass” checkbox during NetPoint setup. For details about setting up the COREid System, see the *NetPoint 7.0 Installation Guide*.
- **Manually**—For details, see the section on configuring attributes manually in the *NetPoint 7.0 Installation Guide*.

## Target Directory Schemas

Just as you extend the schema of your parent virtual directory, you must also extend the native schema used by the LDAP directories included in your virtual directory. You achieve this with the `ldapmodify.exe` utility. For details, see “Extending Directory Schemas” on page 665.

---

**Note:** The NetPoint attributes must also be added to the primary data stores in any split profile included in your virtual directory. For details, see Table 45, “Schema Requirements for Components of a NetPoint-VDE Integration,” on page 636

---

## About Adding Attributes to Target Database Tables

For any databases included in your virtual directory, you must add table columns to simulate the NetPoint auxiliary attributes that enable NetPoint features you plan to use. This applies only to database tables used as primary data sources. For example, you add an Out of Office Indicator column to enable the Surrogate feature in NetPoint workflows.

---

**Note:** SQL-compliant databases do not possess any means to implement LDAP object classes directly. However, you can simulate an object class by mapping, for example, all the rows (user accounts) in a primary database table to the person object class used by the virtual directory.

---

For a listing of the NetPoint auxiliary attributes that enable specific NetPoint functions, see the following:

- Table 49, “Extended Attributes Required by User Manager Functions,” on page 701
- Table 50, “Extended Attributes Required by Group Manager Functions,” on page 702

## Customer Schemas

Optionally, you can further extend the default VDE schema by adding customer attributes from your native data stores. For example, the default VDE person object class is UID, but you can add gensiteorgperson, then specify gensiteorgperson as the Person object class when you run NetPoint setup.

For details on modifying the default VDE schema using the DME interface, see the section on schema configuration in the chapter on configurations and settings in the *VDE 3.0 Product Manual*.

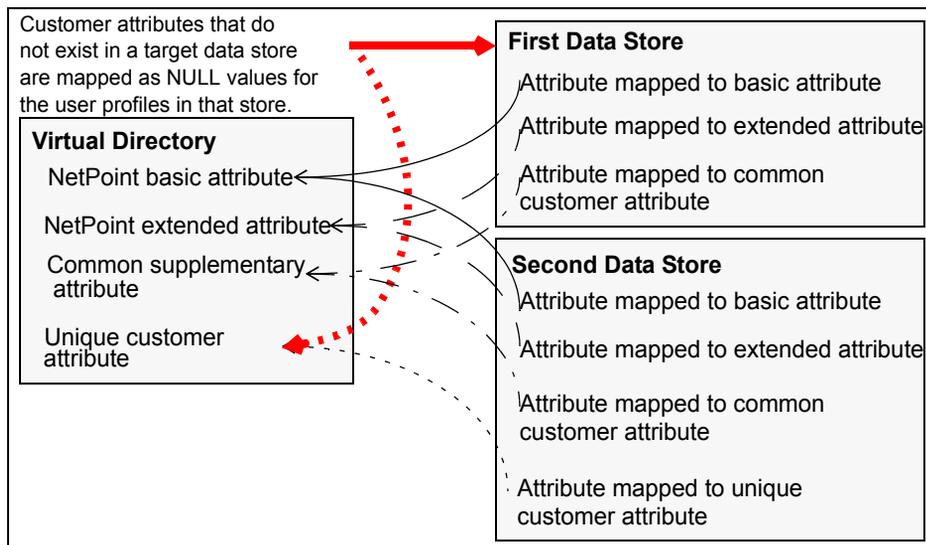
For details on specifying the person object class, see the section about NetPoint object classes in the chapter on setting up the COREid System in the *NetPoint 7.0 Installation Guide*.

---

**Note:** When a customer attribute exists on one target data store, but not in the others, VDE returns the user profiles from those other data stores with that supplementary attribute set to NULL. Figure 37 illustrates this situation.

---

**Figure 37** Mapping Supplementary Attributes to a Simple Virtual Directory



## Integration Scenarios and Limitations

This discussion introduces the three scenarios that are supported when integrating NetPoint with VDE. The following sections also explain the limitations you encounter with each scenario.

- “Heterogeneous LDAP Directories” on page 640
- “Multiple RDBMS Databases” on page 641
- “Split-Profiles” on page 643

# Heterogeneous LDAP Directories

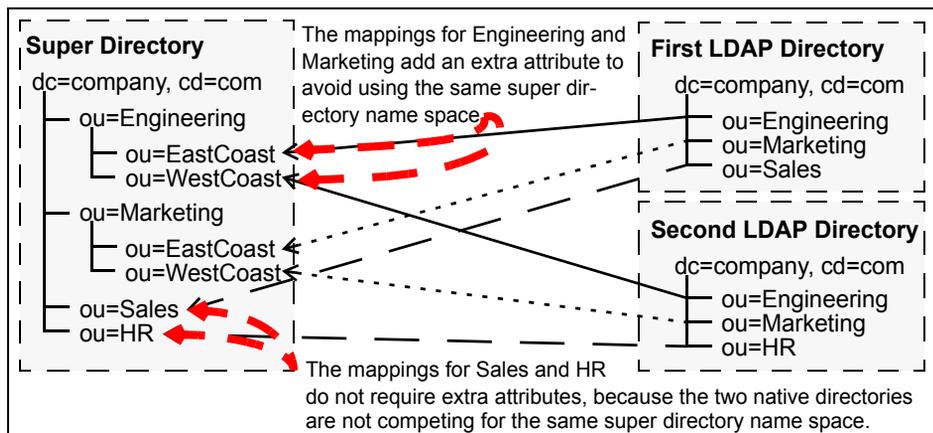
The NetPoint-VDE integration can connect to multiple LDAP v3 directories from one or more vendors. Each directory can use a different schema (attributes and object classes). Because VDE transforms data at run-time, NetPoint sees the aggregated directories as a single directory to which the NetPoint schema has been uniformly applied.

When NetPoint is connected to a VDE system that includes just LDAP directories (and no RDBMS data stores), all Access System and Identity System functionality is available. However, you should observe several restrictions when combining directories.

## Restrictions

- NetPoint supports only a single Person object class and a single Group object class associated with each user profile. Therefore, the various (and possibly multiple) Person and Group object classes in the native directories must be mapped to just one Person object class and one Group object class in the virtual directory.
- The native name spaces of the constituent directories can be identical, but those namespaces must map to different namespaces within the virtual directory.
- The login IDs for all users supported by the virtual directory must be unique across all the included directories. Figure 38 illustrates this situation:

**Figure 38** Mapping Identical Namespaces into a Simple Super Directory



- All the attributes mapped from the constituent directories to a given attribute in the virtual directory must use a common data type. For example, the ObOutOfOfficeIndicator attribute cannot be a binary value in one data source, but a date (indicating when the user will return) in another.

- If a native directory enforces referential integrity, references such as Manager or Group Member can only come from the same native directory. If the native directory doesn't enforce referential integrity, and that native directory also supports external references, references can reside in other directories.
- RDN (relative distinguished name) is not supported.

## Multiple RDBMS Databases

Databases that include a single table that contributes all the user profile attributes mapped by VDE *can* be federated into the virtual directory and made visible to NetPoint. In situations where more than one data table contributes attributes to a given user profile, four options exist for joining the tables so as to create a virtual data source containing the complete set of user attributes.

For more details about these options and the limitations each entails, see “About Joining Database Tables in an Embedded Virtual Data Source” on page 641. See also “Database Connectivity Tips” on page 719.

### About Joining Database Tables in an Embedded Virtual Data Source

When more than one data table contributes attributes to a given user profile, four options exist for joining the tables to create a virtual data source containing the complete set of user attributes:

- The native RDBMS Join feature
- The native RDBMS View feature
- The VDE Join View adapter (split profile)
- A custom joiner based on your preferences

Figure 39 illustrates the four methods for joining multiple database tables within one of the three supported types of embedded virtual data sources.

**Figure 39** Methods for Joining Tables within a Virtual Directory

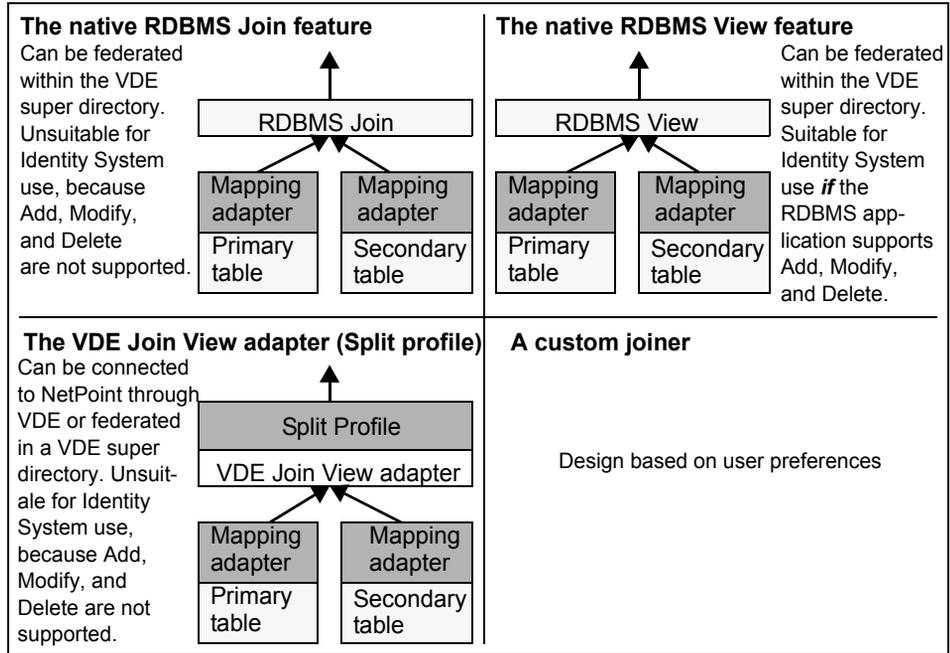


Table 46 lists the specific limitations associated with each method:

**Table 46** Methods for Joining Database Tables within a Virtual Directory

Method	Suitability and Limitations
<p><b>The native Join feature of the host RDBMS application</b></p> <p>The resulting Join is then federated as part of the virtual directory</p>	<ul style="list-style-type: none"> <li>• Suitable for Access System use, because NetPoint Read and Search operations are both supported.</li> <li>• Not suitable for Identity System use, because NetPoint Add, Modify and Delete operations are not supported.</li> </ul>
<p><b>The native View feature of the host RDBMS application</b></p> <p>The resulting View is then federated as part of the virtual directory.</p>	<ul style="list-style-type: none"> <li>• Suitable for Access System use, because NetPoint Add and Search operations are supported.</li> <li>• Suitable for Identity System use <i>only</i> if the native View feature of the RDBMS application supports Add, Modify, and Delete operations.</li> </ul>

**Table 46** Methods for Joining Database Tables within a Virtual Directory

Method	Suitability and Limitations
<p><b>The VDE Join View adapter method</b></p> <ul style="list-style-type: none"><li>• Each data source connects to the Join View adapter through an RDBMS adapter.</li><li>• The result is a split profile, which can be connected to NetPoint through VDE or be federated as part of the virtual directory, which is then connected to NetPoint.</li></ul>	<ul style="list-style-type: none"><li>• Suitable for Access System use, because NetPoint Add and Search operations are supported.</li><li>• Suitable for Identity System use, but Subtype Search, Add, and Delete operations can be performed only on the primary table.</li><li>• All limitations to LDAP directories joined with the Join View adapter also apply to databases joined with the Join View adapter. For details, see “Join View Adapter Requirements and Limitations” on page 643.</li></ul>
<p><b>A custom joiner method</b></p>	<p>You can write a custom joiner to overcome the limitations imposed by the standard Join View adapter or the native Join and View features of your RDBMS application.</p> <p>This involves custom programming. For details, consult the section on joiners in Chapter 9 of the <i>VDE 3.0 Product Manual</i>.</p>

## Split-Profiles

Virtual directories which draw the attributes for each user profile from two or more data sources are known as *split profiles*. These data sources can include any combination of LDAP directories and relational databases.

One data store serves as the *primary data source*. The schema of this data store must be extended with the NetPoint-specific user data. For details, see “About Schema Extension” on page 635.

All additional data stores are *secondary data sources*. Not all Identity System functionality is supported for these secondary data stores. See “Join View Adapter Requirements and Limitations” on page 643. It is not necessary to extend the NetPoint user schema to secondary data stores.

You can join the data sources in a split profile either through the standard VDE Join View tool (recommended method), or through a custom joiner. For details on creating a custom joiner, consult the *VDE 3.0 Product Manual*.

## Join View Adapter Requirements and Limitations

The Join View adapter supports all Access System operations on attributes that reside in either the primary or secondary data stores. This includes authentication, authorization, auditing, and single sign-on.

---

**Note:** NetPoint Identity System operations are supported, with the following restrictions.

---

## Restrictions

- The user login ID attribute for the split directory must reside in the primary data store. The user login password and the user full name attribute must also reside in the primary data store
- The NetPoint user schema must reside the in the primary data store.
- Base-level searches are supported for both the primary and secondary data stores.
- Sub-tree searches are supported only for the primary data store. By implication, the following restrictions apply:
  - Attributes residing in a secondary data store must not be configured as searchable.
  - Attributes residing in a secondary data store must not be configured for filter operations involving sub-tree searches. This includes:
    - domain filters
    - dynamic group filters
    - group subscription filters
    - Query Builder filters
- Modify is supported for all attributes, without regard to the specific data store in which those attributes reside.
- Users, groups, and other objects can be created only in the primary data store.
- Only users, groups, and other objects in the primary data store can be deleted. (NetPoint applications cannot delete objects in the secondary data store. This can be accomplished only through the target RDBMS application or LDAP directory, which may lead to synchronization problems in real-time environments.)
- A given attribute can be configured from only one data store. Join values are not supported.

## Integration Requirements

NetPoint connects to the virtual directory just as it connects to any other LDAP directory; therefore, most supported NetPoint configurations should integrate smoothly with VDE. The following sections list support and requirement details for various aspects of the NetPoint-VDE Integration.

**NetPoint**—The LDAP directory branches containing NetPoint configuration and policy data must reside on one or more directory servers native to the NetPoint system. In other words, the configuration and policy branches cannot reside

anywhere in the virtual directory, which contains any and all user data visible to NetPoint.

You specify the locations of your configuration, policy, and user data during installation and setup of each NetPoint component. For details, see the *NetPoint 7.0 Installation Guide*.

**VDE**—NetPoint supports integration with VDE v3.0.1

**Operating System**—You can integrate VDE with NetPoint components installed on host machines running any of the operating systems for which NetPoint provides support.

The LDAP directories and RDBMS databases supported by your virtual directory can be installed on any of the host platforms supported by VDE.

Table 47 lists the operating systems supported for the host machine on which you install VDE:

**Table 47** Host Operating Systems Supported for VDE Installations

Vendor	Operating System Version
Sun	Solaris 8, Solaris 9
Microsoft	Windows 2000 Advanced Server, SP4 Windows Server 2003, Enterprise Edition
RedHat	Linux Enterprise Edition, v2.1
IBM	AIX, v5.2

For the latest support details, see the Web site below:

[https://customers.oblix.com/shared/prodinfo/prod\\_roadmap.cfm](https://customers.oblix.com/shared/prodinfo/prod_roadmap.cfm)

**Java Runtime Environment**—The host machine on which you install VDE must have the Java Runtime Environment installed.

The NetPoint-VDE integration has been tested with JRE v1.4.

**JNDI Driver**—Use the JNDI driver that comes with the VDE v3.0.1 installation package.

**JDBC Driver**—On the machine that hosts VDE, you must install a version of the JDBC driver appropriate for the RDBMS application you connect to the virtual directory. You can obtain the proper driver from the vendor of your RDBMS application or from the following website:

<http://www.octetstring.com/support/JDBC-drivers.php>

If your virtual directory includes databases from multiple vendors, you must install a JDBC driver for each vendor represented. See the *VDE 3.0 Product Manual* for details.

**Data Set**—The NetPoint-VDE integration uses the UTF-8 character set. VDE is localization ready, but only English is supported explicitly.

**Relational Database**—In general, NetPoint can connect to any virtual directory that includes RDBMS databases supported by VDE 3.0.1. Figure 27 on page 625 lists the RDBMS applications specifically supported for the NetPoint-VDE integration. See also “Database Connectivity Tips” on page 719.

**Directory Server**—In general, NetPoint can connect to any LDAP directory server supported by VDE 3.0.1. Figure 26 on page 625 lists the LDAP directory servers specifically supported for the NetPoint-VDE integration.

**Caching**—VDE does not provide explicit caching support.

For additional details on NetPoint-VDE integration requirements and support, see:

- “Security Connection Support” on page 646
- “Authentication Support” on page 647
- “Access Control Support” on page 647
- “Failover Support” on page 648

## Security Connection Support

Open or SSL connections are supported between NetPoint and VDE and between VDE and the native data stores.

For Active Directory, ADSI is not supported. You must use SSL, if you want to change passwords within your Active Directory. Otherwise, you can use Open mode.

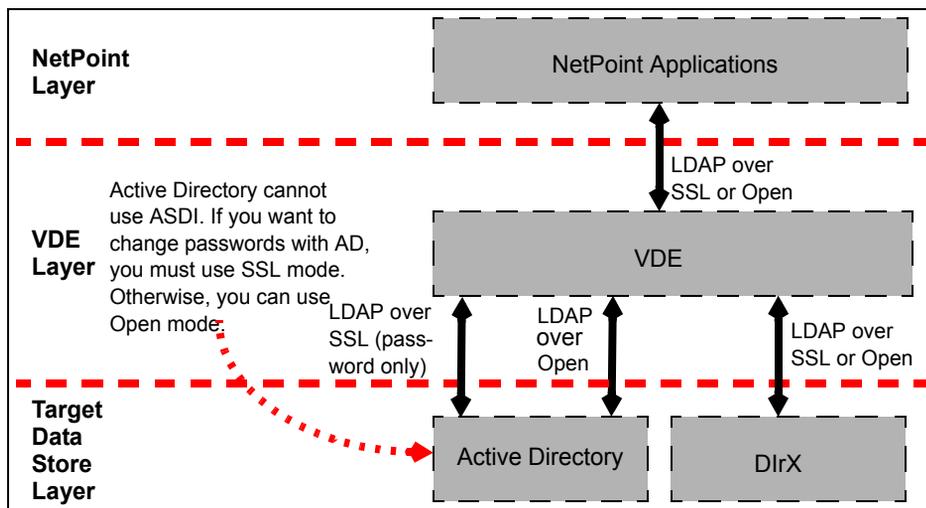
---

**Note:** For best performance when connecting an Active Directory to VDE, specify Password Only SSL as the security connection mode. For this scenario, you will also need to create an Open connection between VDE and the Active Directory.

---

Figure 40 illustrates the protocols used by the connections within the NetPoint-VDE integration.

**Figure 40** Protocol support for a simple NetPoint-VDE Integration



## Authentication Support

VDE supports the following authentication methods:

- Pass credential authentication
- Pure proxy

## About Pass Credential Authentication

If you use Pass Credential authentication for your NetPoint-VDE integration, you must set Pass Credentials to “Always” (or Bind Only) to ensure that VDE passes the user distinguished name and password supplied by NetPoint to the proxied LDAP directory.

For background details, consult the section on directory namespace and attribute mapping in the chapter covering configurations and settings in the *VDE 3.0 Product Manual*.

## Access Control Support

Make sure that both NetPoint and VDE access control are turned on and the default settings are in effect for the connection between NetPoint and VDE. For background details, consult the chapter on security and access control in the *VDE 3.0 Product Manual*.

For the connections between VDE and the target data stores, turn on the access control supported each target data store. (Because VDE is an LDAP client, it must use the access control implementation native to each target directory server). For

details, consult the section on access control and the LDAP adapter in the configuration and settings chapter of the *VDE 3.0 Product Manual*.

## Failover Support

The NetPoint-VDE integration implements failover support using the existing failover capabilities in the NetPoint, VDE, directory server, and RDBMS applications. You can implement failover on the following three levels:

- NetPoint failover
- VDE target source failover
- Target data store failover

**NetPoint Failover**—An Identity or Access Server can connect to one or more primary virtual directory instances and one or more secondary VDE instances.

- See the section on adding database instances to an LDAP server profile in the chapter on Managing and Configuring the COREid System in the *NetPoint 7.0 Administration Guide Volume 1*.
- See the *NetPoint 7.0 Deployment Guide* for details about configuring failover in NetPoint.
- See the section on fault-tolerant deployments in the chapter on virtual directory planning in the *VDE 3.0 Product Manual*.

**VDE Target Source Failover**—VDE can implement failover protection between your virtual directory and your target data stores. For details, see the section on fault-tolerant deployments in the chapter on virtual directory planning in the *VDE 3.0 Product Manual*.

**Target Data Store Failover**—Often, RDBMS applications and LDAP directory servers support failover in the form of clustering at the target data store level. In general, the mechanisms that implement this capability operate automatically and are not visible to VDE or NetPoint. For details, consult the documentation for your RDBMS application or LDAP directory server.

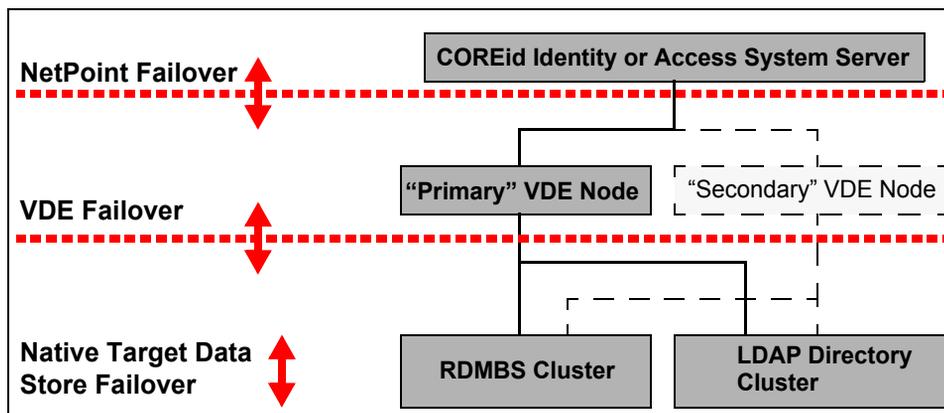
---

**Note:** This chapter does not provide any specific procedures for configuring failover for your environment. You can set up failover as you usually do, according to your product documentation.

---

Figure 41 illustrates the types of failover potentially available within a NetPoint-VDE integration.

**Figure 41** Failover Options for NetPoint-VDE Integrations



## About the NetPoint-VDE Integration Process

The following topics introduce two different situations. Be sure to choose the topic that corresponds to your environment:

- **NetPoint Not Installed**—See “Integrating VDE when Installing NetPoint” on page 649
- **NetPoint Installed**—See “Integrating VDE with Existing NetPoint Installations” on page 650

### Integrating VDE when Installing NetPoint

When you have *not* yet installed NetPoint, you need to complete the activities outlined below to complete the integration between VDE and NetPoint.

#### **Task overview: Installing and integrating VDE and NetPoint includes**

1. “Preparing Your Environment” on page 652
2. “Installing and Configuring VDE and DME” on page 657
3. “Installing the First NetPoint COREid Server” on page 663
4. “Extending Directory Schemas” on page 665
5. “Creating Mapping Files for Adapters” on page 667
6. “Creating Data Store Adapters” on page 669
7. “Customizing Adapters and Mapping Files” on page 679
8. “Completing NetPoint Installation and Setup” on page 698

9. “Testing Your Integration” on page 699

## Integrating VDE with Existing NetPoint Installations

When you *have* a working NetPoint installation (version 5.2.x or 6.x) with a native directory or RDBMS, you can integrate VDE to provide access to the originally installed native directory server and additional user directories and databases.

In this case, you complete activities in the task below to upgrade your NetPoint installation, prepare VDE, extend directory schemas, then reconfigure NetPoint to use the COREid Data Anywhere option as a *user data* directory server.

---

**Important:** The DNConversion Toolkit is a stand-alone package that you *must* download from the Oblix Web site to ensure that you have the latest version. *Always* use files in the stand-alone DNConversion Toolkit. An *earlier* version of the toolkit is included in the *COREid\_702\_install\_dir*. However, do *not* use the toolkit in the *COREid\_install\_dir*.

---

### Task overview: Integrating VDE with existing NetPoint installations

1. In your existing NetPoint installation, confirm that user data and Oblix (NetPoint configuration) data are stored on *separate* directory servers.
2. Download the latest version of the DataAnywhere Toolkit when you download the NetPoint 7.0.2 installation packages, then unzip/untar the files as usual.  
*DNConversionTool\_install\_dir\oblix\tools\DataAnyWhere*
3. Upgrade your existing NetPoint installation with the native directory to NetPoint 7.0.2 using instructions in the *NetPoint 7.0 Upgrade Guide* and the NetPoint 7.0.2 installation packages.
4. Prepare any additional directories and databases to be integrated with VDE, as described in:
  - “Preparing Your Environment” on page 652
  - “Extending Directory Schemas” on page 665
5. Install and configure VDE/DME, as described in the *VDE 3.0 Installation Guide* and in “Installing and Configuring VDE and DME” on page 657.
6. Use the Oblix-provided DN conversion tool to clean up existing NetPoint configuration and policy data in the native directory, as indicated below:
  - a) Locate the conversion tool in:

*DNConversionTool\_install\_dir\oblix\tools\DataAnyWhere\conversion\_tools\obmigrateDN.exe*

b) **Same Object Class**—If the object class for the new VDE virtual directory remains the same as the object class used by the original NetPoint installation, *and* the only change involves the namespace hierarchy, the Oblix tree (NetPoint configuration and policy data) can be *patched* using the DNConversion.exe tool, which:

- Saves the original configurations
- Converts the native DN used by the existing NetPoint installation to the new *virtual* (logical) DN used by the VDE virtual directory. DN references such as the UIDs in the policy and the NetPoint Administrators, Master Identity and Access Administrators, and directory server administrators need to be converted from the old native DN to the new virtual DN.

**Note:** The configuration file DNConversion.xml provides connection information to the NetPoint configuration (Oblix data) branch, as well as the attributes to be converted. For details, see “Contents of the Data Anywhere Toolkit for Integrating NetPoint and VDE” on page 704.

c) **Different Object Class**—If the virtual object class that VDE will make visible to NetPoint *differs* from the virtual object class used by your original (pre-VDE) NetPoint installation, you might need to reconfigure your entire NetPoint installation, during which you typically:

- Remove the entire Oblix tree (the NetPoint configuration and policy branches of the LDAP DIT).
- When you complete step 7 below to manually complete a new COREid System setup procedure be aware that you will *lose* any configured policies and the like.

7. **All**—Re-run setup manually for the NetPoint COREid System using the following specifications, as needed:

- a) **User Data Directory Server**—Select COREid Data Anywhere as the directory type.
- b) **User Data ... Host**—Specify the machine hosting VDE.
- c) **User Data ... Port**—Specify the VDE LDAP licensing port.
- d) **Searchbase**—Specify the VDE virtual DN, which may be any part of the virtual tree.
- e) **Oblix Data Directory Server**—Select the native directory, as you normally do, because NetPoint configuration and policy data must be stored outside the VDE virtual directory, which contains user data only, as far as NetPoint is concerned.
- f) **Automatically Update Schema**—Select Yes to automatically update the VDE schema with NetPoint auxiliary attributes.

- g) **Automatically Configure Person and Group Object Classes**—Choose Yes or No, as you normally do, to configure the VDE schema.

**Note:** For details about manually configuring Person and Group object classes, see the *NetPoint 7.0 Installation Guide*.

8. Finish setup as you normally do.

See the *NetPoint 7.0 Administration Guide Volume 1* for additional information about re-running NetPoint setup manually.

## Preparing Your Environment

Preparing your environment so that NetPoint can integrate with VDE (also known as COREid Data Anywhere), includes the following activities.

### **Task overview: Preparing your environment includes**

1. Download the DataAnywhere Toolkit when you download the NetPoint 7.0.2 installation packages.
2. “Identifying Factors for Designing Your Integration” on page 652.
3. “Preparing Directory Servers for Integration” on page 655
4. “Preparing Relational Databases for Integration” on page 656

## Identifying Factors for Designing Your Integration

Before you start the integration, you need to collect information and make decisions to guide the design of your NetPoint-VDE integration.

Consider and answer the following questions, performing background investigation, as necessary.

### **To identify factors for this integration**

1. Determine the data stores do you want to access through VDE.

You can federate LDAP directories and RDBMS databases within your virtual directory. You can also create and federate embedded virtual data sources such as split profiles, native RDBMS Joins, and native RDBMS Views.

**Qualifying LDAP Directories as Target Data Sources**—The incorporation of LDAP directories is relatively straightforward, because NetPoint supplies both an adapter template and a schema mapping template for each of the LDAP directory servers NetPoint supports for VDE integration. See “About the Data Anywhere Toolkit” on page 703.

The only major restriction is that two directories cannot occupy the same namespace in a super directory, but you can prevent this sort of collision by mapping the namespaces used by the native directories to unique namespaces within the super directory. For details, see “Aggregated Namespaces” on page 628.

**Qualifying RDBMS Databases as Target Data Sources**—For RDBMS databases, you must first determine whether all the essential information NetPoint needs to operate exists within a single table. This includes the database columns that are mapped to the following attributes:

- UID (user login id)
- User password
- Full Name
- Person object class, which is generally implicit through the name of the table in which the essential fields reside. For example, all the user accounts in the Employee table of a database can be mapped to the inetorgperson person object class in the virtual directory. If you have another table such as Consultants, you can also map all of its user accounts to inetorgperson, then use the native RDBMS Join or View features to concatenate the user accounts. The important principle to observe is that all user accounts must be associated with the single person object class specified by the virtual directory.
- Whatever NetPoint user branch attributes are necessary to enable the specific NetPoint features you plan to use. For instance, you can add a column labelled OOO (Out of Office) to the Employee table so that you can run workflows against the virtual directory.

If all essential information exists within a single table, you can federate the database as part of the virtual directory. If not all the essential information exists in the database, or that information is spread across more than one table, the database might not be suitable for inclusion in the virtual directory, or you may have to use one of three available methods to transform the database into an embedded virtual data source, which you then federate within the virtual directory.

2. Determine the items of virtual directory information you want to make visible to NetPoint.

To ensure that NetPoint can interact with the virtual directory, you must make the essential items listed in step 1 visible to NetPoint. You should

also determine what customer attributes you want to make visible to NetPoint. For instance, you can make employee cell phone numbers or birthdays visible to NetPoint by adding those attributes to the virtual directory.

**3.** Determine the best approach to mapping the object class and attributes.

This depends on the native schema used by your target data stores. You are free to create virtually any schema you wish VDE to make visible to NetPoint, but you should keep in mind the following points:

- Information from two different target data stores can never occupy the same namespace in a super directory
- If your virtual directory includes any embedded virtual data stores, you should avoid workflows that create, delete, or otherwise modify user accounts, because you generally cannot change information in the secondary data stores
- The secondary data stores in embedded virtual directories (split profiles, RDBMS Joins, and RDBMS Views) cannot be included in filters or sub-searches. In other words, embedded virtual data stores such as split profiles, RDBMS Joins, and RDBMS Views are suitable for Access system operations (authentication and authorization), but they are generally inappropriate for identity management operations, because key functionality such as Add, Delete, and Modify are often not available for data in the secondary data stores.
- The NetPoint-VDE integration does not simultaneously support attributes with multiple values and databases used as target data stores. You can use multi-valued attributes if your virtual directory contains LDAP directories exclusively.

**Note:** If you must use multi-valued attributes in conjunction with a database, see “NetPoint-VDE Integration Templates” on page 708 and “Multi-Value Attribute Problems” on page 721.

**4.** Decide what operations you want NetPoint to perform on each piece of information in the virtual directory.

If you want to use certain NetPoint features such as surrogates in workflows, then you have to add columns to the primary database tables in your embedded virtual data stores. For lists that correlate specific NetPoint User Manager and Group Manager functions to auxiliary NetPoint user attributes which that must be added to the primary tables in databases serving as target data stores, see “About Adding Attributes to Target Database Tables” on page 638 and “NetPoint Auxiliary Attributes” on page 700.

5. From the standpoint of policy management and identity management, what is the optimal DIT hierarchy for your virtual directory?

You can choose between disjoint searchbases or a unified searchbase. For details, see “About Searchbase Options” on page 626.

If you chose the super directory option, you can optimize namespace aggregation and schema mapping to fit the needs of your organization. For instance, the engineering directories of two companies can be aggregated following a corporate merger so that only one set of access policies has to be configured for all the engineers in the new corporation. See “Aggregated Namespaces” on page 628 for an simple example of namespace mapping that handles such a scenario.

6. Decide what machines will host integration components and where to install:

- NetPoint
- VDE
- RDBMS applications
- Native LDAP directory servers

7. What types of secure connections will link the various components?

See “Security Connection Support” on page 646 for details.

8. Continue with:

- “Preparing Directory Servers for Integration” on page 655
- “Preparing Relational Databases for Integration” on page 656

## Preparing Directory Servers for Integration

You need to install and configure any native directory servers that you plan to integrate into the VDE virtual directory. This you can accomplish now.

---

**Note:** A second requirement, which you will perform later, is to extend the native schema of each back-end directory server with NetPoint-related user and group information so your VDE and native schemas include the same NetPoint attributes.

---

### To prepare each directory server

1. Review “Integration Requirements” on page 644.
2. Install back-end directory servers according to vendor instructions.  
Later you will extend the native schema with NetPoint-related attributes.
3. Proceed as follows, depending on your environment:
  - “Preparing Relational Databases for Integration” on page 656
  - “Installing and Configuring VDE and DME” on page 657

## Preparing Relational Databases for Integration

Before you continue, you must begin preparing each of your relational databases for inclusion in the directory. This procedure is a prerequisite for creating an RDBMS-specific adapter.

### To prepare each relational database for integration

1. Install and configure your RDBMS according to vendor instructions.
2. Verify that the database contains, in a single table, all the fields that must be mapped to the essential attributes in the NetPoint schema used by the virtual directory, which are the following:
  - UID
  - User Password
  - Full Name
3. Consider the following:
  - If the database does *not* contain all essential fields, it may not be suitable for inclusion in the virtual directory.
  - If all the essential fields are not in the same table, you cannot include that table in the virtual directory (because the essential fields residing in secondary tables are not searchable). Optional, customer fields can reside the secondary tables.
  - If all the essential fields are in the same table, you might have to create an embedded virtual data store using one of the following methods:
    - The Join View adapter
    - The View feature native to your RDBMS application
    - The Join feature of your RDBMS application

**Note:** The three methods for incorporating multi-table databases mentioned above necessarily limit certain NetPoint Identity System functionality.
4. Consult the following Web site for versions of the JDBC driver and additional JDBC-related information:  
<http://www.octetstring.com/support/JDBC-drivers.php>
5. From the Web site of your RDBMS application vendor, download the necessary driver libraries.  
  
For example, for Oracle you need to download the Oracle JDBC thin driver: `ojdbc14.jar` (or newer).
6. Install the driver according to your vendor instructions.

7. Skip to “Installing and Configuring VDE and DME” on page 657.

Later you will be instructed to deploy the JDBC driver for your target database on the machine hosting VDE.

See also “Database Connectivity Tips” on page 719.

## Installing and Configuring VDE and DME

After you have selected an integration configuration and prepared your data sources for integration, you must install both the NetPoint and VDE software necessary for the integration. The following task overview summarizes the procedures you need to complete.

### **Task overview: Installing and configuring VDE and DME includes**

1. “Installing VDE” on page 657.
2. “Installing DME” on page 658.
3. “Creating a Project Space and Server” on page 658
4. “Obtaining/Updating Sample Adapter and Mapping Templates” on page 659
5. **RDBMS**—“Deploying JDBC Driver Libraries for Your RDBMS” on page 660
6. “Configuring the VDE SSL Listener (Optional)” on page 661

## Installing VDE

You install the VDE as usual. There are no specific measures you need to take to facilitate integration with NetPoint.

### **To install VDE**

1. Install and setup VDE following the instructions in the *VDE 3.0 Installation Guide*.
2. Use the default settings provided in the VDE documentation.
3. Record information about your VDE installation for use when you install the first NetPoint COREid Server, including:
  - **Host Name**—The DNS hostname of the machine hosting VDE.
  - **Port Number**—The VDE LDAP licensing port.
  - **Bind DN** for the user data directory server—The VDE virtual DN, which may be any part of the virtual tree.
  - **Password**—The password for the user data bind DN.

4. Configure VDE as described in your VDE documentation.

---

**Note:** If you want to configure an SSL connection between the VDE and NetPoint, see “Configuring the VDE SSL Listener (Optional)” on page 661.

---

5. Proceed with “Installing DME” on page 658.

## Installing DME

You install the DME as usual. There are no specific measures you need to take to facilitate integration with NetPoint.

### To install the DME

1. Follow the instructions in the *VDE 3.0 Installation Guide*.
2. Use the default settings provided in your VDE documentation.
3. Configure the DME as described in your VDE documentation.
4. Proceed with “Creating a Project Space and Server” on page 658.

## Creating a Project Space and Server

You create a project space and server using the DME, as you usually do. Some sample steps are provided here; however, these are not intended as a complete tutorial.

For additional information, see your VDE documentation.

### To create a project space and server

1. Start > DME.
2. From the menu under the Server Navigator window, select Directory Management Project.
3. Specify a unique project name.
4. Right-click the project name, then select New > Server.
5. Enter a unique server name.
6. Click Finish.
7. Proceed to “Obtaining/Updating Sample Adapter and Mapping Templates” on page 659.

## Obtaining/Updating Sample Adapter and Mapping Templates

You need to complete the procedure below to ensure that the appropriate Oblix-provided sample templates are available in the Adapter Template list within VDE. You can create your own templates from scratch if you like. Oblix provides two types of sample templates specific to each data store and also to a specific *user-defined* schema:

**Sample Adapter Templates**—Sample templates for vendor-specific adapter files that you can use as the basis for individual adapters that connect native data stores to VDE. The Oblix-provided sample adapter template files are stored in:

```
DNConversionTool_install_dir\oblix\tools\DataAnywhere\adapter_templates
    \ldap
    \database
```

---

**Important:** The DNConversion Toolkit is a stand-alone package that you *must* download from the Oblix Web site to ensure that you have the latest version. Do *not* use the toolkit in the *COREid\_install\_dir*.

---

**Sample Mapping Templates**—Sample Mapping files that you can use so that VDE can transform the schema (or database fields) used by native data stores to the logical schema used by the aggregated virtual directory made visible to NetPoint. Oblix-provided sample mapping templates are provided in:

```
DNConversionTool_install_dir\oblix\oblix\tools\DataAnywhere
    \mapping_templates
```

---

**Note:** At this point you just obtain and update the sample templates. Later, you use these to configure each connector and perform the schema and the namespace mapping. For more information, see “Creating Mapping Files for Adapters” on page 667 and “Creating Data Store Adapters” on page 669 and “Customizing Adapters and Mapping Files” on page 679.

---

### To copy the sample templates to the DME

1. Obtain and update Oblix templates using the step below that is appropriate for your environment.
2. **No Oblix Samples Yet**—In the DME, Click Help > Software Updates > Find and Install.
3. **Automatic Update, Oblix Samples Installed**—Complete the steps below based on your installation:
  - a) Choose “Search for updates of currently installed features”.
  - b) Choose “Search for new features to install”.

4. **Manual Update**—Complete the steps below to manually update Oblix sample templates.
  - a) Unzip the distributed zip file (for example, COREidFeatures\_7.0.2.bin.dist.zip) to the following DME directory:  
*DME\_install\_dir*\eclipse  
for example, C:\Program Files\OctetString\DME\eclipse
  - b) Restart the DME.
5. Proceed to the appropriate discussion below, depending on your environment:
  - “Deploying JDBC Driver Libraries for Your RDBMS” on page 660
  - “Configuring the VDE SSL Listener (Optional)” on page 661
  - “Installing the First NetPoint COREid Server” on page 663

## Deploying JDBC Driver Libraries for Your RDBMS

You complete this procedure only if your integration will include an RDBMS.

After downloading and installing the JDBC driver libraries and completing the steps above, you need to complete the procedure below to deploy each library. For example, for Oracle you need to deploy the Oracle JDBC thin driver: ojdbc14.jar (or newer).

Again, the instructions here give you an idea of how to proceed. For more information, see your VDE product manual.

### To deploy a JDBC driver library

1. Complete activities in “Preparing Relational Databases for Integration” on page 656.
2. Launch the DME, as usual, and navigate to the Server Navigator window.
3. Right click on VDE Server, then select Manage Server Libraries from the menu.
4. Select a file.
5. Select New > Deploy.

The JDBC Driver files are stored in *JDBC\_Driver\_install\_dir*/lib (for example, C:\Program Files\JDBC).

6. Select the libraries for your environment.
  - msbase.jar
  - mssqlserver.jar
  - msutil.jar
7. Deploy as usual.

## Configuring the VDE SSL Listener (Optional)

The procedure below is required *only* if you wish to set up an SSL connection between NetPoint and the VDE. Skip this section if you plan to use an Open connection between NetPoint and the VDE.

### To configure the VDE SSL listener

1. Generate a private key, as indicated below:
  - a) Right-click *server* and select Server-Manager Server keys.
  - b) Click Generate Key.
  - c) Fill in the key information.

The Common Name you use must be exactly the host name you use in COREid later on.
2. Generate a certificate request, as follows:
  - a) Select the key just generated from Key/Certificate window
  - b) Click Request Certificate.
3. Sign the certificate request, as follows:
  - a) Start MicroSoft certificate service using <http://machine/certsrv>
  - b) Click the link Request a Certificate
  - c) Click the link Advanced Certificate Request
  - d) Click the link Submit a Certificate Request by Using Base 64....
  - e) In an editor, open the certificate request file generated in step 2.
  - f) Copy the text and past it to the Base 64-encoded window in the certificate service.
  - g) In Certificate Template, select Web Server and then Submit.
  - h) Download the CA certificate in Base 64 encoded format.
4. Import the signed certificate to VDE, as described below:
  - a) On DME Key/Certificate window, click Import.
  - b) Select the certificate file obtained in step 3.
  - c) Provide the alias exactly the same as the alias given for the key in step 1.
  - d) Once you Finish, you should see the Issuer of the key entry is updated to the CA
5. Configure LDAP listener with SSL, as described next:
  - a) In DME Server Navigation Pane, right click on Listeners and select New – Ldap Listener

- b) Provide a port
  - c) In Server Key Alias, select the key entry created in 4.
  - d) Save to Server.
6. Install the certificate in NetPoint, according to the conditions below:
- **NetPoint Not Installed**—If you have not yet installed NetPoint you can install the certificate automatically during COREid Server installation. In this case, skip to “Installing the First NetPoint COREid Server” on page 663.
  - **NetPoint Installed**—If you have already installed NetPoint, complete the steps below.
7. Create the cert8.db, if needed:
- a) Navigate to `COREid_install_dir\identity\oblix\tools\certutil`
  - b) Issue the command below:  

```
certutil -d COREid_install_dir\identity\oblix\config -N -f
```
8. Import the root CA to NetPoint using the command below:  

```
certutil -d COREid_install_dir\identity\oblix\config -A -n ldap -a -t -C -i root_ca_file
```
9. Reconfigure the COREid Server as follows:
- a) Navigate to the COREid System Console > System Admin > System Configuration > Configure Directory Options
  - b) Locate the user profile and DB instance, as described in the *NetPoint 7.0 Administration Guide Volume 1*.
  - c) Mark SSL, then enter the secure port of VDE.
  - d) Restart the COREid Server.
  - e) Repeat this for all the instances for which you want to use SSL.
  - f) Rerun NetPoint setup manually, as described in the *NetPoint 7.0 Administration Guide Volume 1*.

# Installing the First NetPoint COREid Server

For successful integration with VDE, you must complete NetPoint installation in stages. During this first phase you install *only* the *first* COREid Server. This installation provides the ldif files you need to extend the native schema of directory servers you plan to integrate with VDE.

During COREid Server installation for VDE, you *must* specify the following:

- **User Data Directory Server**—Select COREid Data Anywhere when prompted for the *user data* directory server.
- **Configuration and Policy Data Directory Server**—Specify a native directory server when prompted for the location of configuration and policy data. The configuration and policy branches *cannot* reside on the same host machine as your VDE installation.

When you finish the procedure below, you can extend the schema of native directory servers you plan to integrate with VDE.

---

**Important:** If you have NetPoint installed with a native directory server, see “Integrating VDE with Existing NetPoint Installations” on page 650, then skip to “Extending Directory Schemas” on page 665.

---

## To install the first COREid Server

1. Review NetPoint installation prerequisites, requirements, options, and COREid Server installation considerations in the *NetPoint 7.0 Installation Guide*.
2. Start installing the first COREid Server, and proceed through defining directory server details, as described in the *NetPoint 7.0 Installation Guide*.

Storing Oblix (NetPoint configuration data) separately is required. Later you are asked to provide details about both the user data directory server and Oblix data directory server.

3. When asked where data is to be stored, indicate that Oblix data is to be stored separately.

Oblix data stored separately

Oblix recommends that you automatically extend the schema during installation of the first COREid Server. You update the schema only once. Either Yes response will result in questions about directory server type and specifications.

4. When asked about updating the schema, select the Automatic schema update option for separate storage of user and Oblix data.

5. When asked about *user data* directory server details, specify the following for this integration:
  - a) **User Data Directory Server**—COREid Data Anywhere.
  - b) User Data Directory Server Details
    - **Host name**—The DNS hostname of the machine hosting VDE.
    - **Port number**—on which the directory server listens—Specify the VDE LDAP licensing port.
    - **Bind DN**—For the *user data* directory server, specify the VDE virtual DN, which may be any part of the virtual tree.
    - **Password**—Specify the password for the *user data* bind DN.
6. When asked about *configuration data* directory server details, specify the following for this integration:
  - a) **Oblix Data Directory Server**—A native directory server type.
  - b) Oblix Data Directory Server Details
    - **Host name**—The DNS hostname of the machine hosting a native directory where you will store NetPoint configuration data.
    - **Port number**—Specify the port on which the NetPoint configuration data directory server listens.
    - **Bind DN**—For the *configuration data* directory server.
    - **Password**—The password of the *configuration data* bind DN.
7. Finish installing the first COREid Server as usual.
8. Proceed to the next topic, “Extending Directory Schemas” on page 665 and continue through all topics.

---

**Important:** Finishing the NetPoint installation and setup is the *last* thing you do to complete this integration. If you do not complete all other activities first, your integration with VDE may not be successful.

---

# Extending Directory Schemas

The NetPoint COREid System requires attributes assigned to the Full Name, Login, and Password semantic types for Person and Group object classes.

Before you continue, you need to complete the following procedure to ensure that you:

- Extend back-end native directory schemas with NetPoint attributes using the appropriate Ldif file in:

```
DN_ConversionTool_install_dir/oblix/tools/DataAnyWhere/  
OblixUserSchema/*.ldif
```

- Extend your VDE schema with NetPoint attributes using the VDE\_user\_schema\_add.ldif file in

```
DN_ConversionTool_install_dir/oblix/tools/DataAnyWhere/  
OblixUserSchema/VDE_user_schema_add.ldif
```

- Extend your VDE schema to represent all your back-end data sources using the appropriate step in the following procedure

---

**Important:** The DNConversion Toolkit is a stand-alone package that you *must* download from the Oblix Web site to ensure that you have the latest version. Do *not* use the toolkit in the *COREid\_install\_dir*.

---

## To extend directory schemas

1. Locate the Ldif files to use when you extend the schema of a back-end directory server you are preparing for inclusion in the virtual directory, as follows:

```
DN_ConversionTool_install_dir/oblix/tools/DataAnyWhere/  
OblixUserSchema/
```

2. Use Table 48 as a guide to manually configure attributes for each specific back-end directory server you will include in the virtual directory.

**Table 48** Files and Commands to Extend Native Schemas with NetPoint Attributes

Directory Server and Idif File	Manual Schema Update Commands
Active Directory ADUserSchema.ldif or ADAuxSchema.ldif depending on your environment	ldifde -s <i>host</i> -t <i>port</i> -a <i>bind-dn</i> -w <i>password</i> -c <i>fromDN toDN</i> -i -f ADUserSchema.ldif
ADAM ADAM_user_schema_add.ldif or ADAMAuxSchema.ldif depending on your environment	ldifde -s <i>host</i> -t <i>port</i> -a <i>bind-dn</i> -w <i>password</i> -c <i>fromDN toDN</i> -i -f ADAM_user_schema_add.ldif
SunONE • iplanet_user_schema_add.ldif • iplanet5_user_index_add.ldif	ldapmodify -h <i>host</i> -p <i>port</i> -D <i>bind-dn</i> -w <i>password</i> -a -f iplanet_user_schema_add.ldif ldapmodify -h <i>host</i> -p <i>port</i> -D <i>bind-dn</i> -w <i>password</i> -a -f iplanet5_user_index_add.ldif
eDirectory • NDS_user_schema_add.ldif • NDS_user_index_add.ldif	ldapmodify -h <i>host</i> -p <i>port</i> -D <i>bind-dn</i> -w <i>password</i> -a -f NDS_user_schema_add.ldif ldapmodify -h <i>host</i> -p <i>port</i> -D <i>bind-dn</i> -w <i>password</i> -a -f NDS_user_index_add.ldif
SecureWay • v3.user.ibm_at.ldif • v3.user.ibm_oc.ldif	ldapmodify -h <i>host</i> -p <i>port</i> -D <i>bind-dn</i> -w <i>password</i> -a -f v3.user.ibm_at.ldif ldapmodify -h <i>host</i> -p <i>port</i> -D <i>bind-dn</i> -w <i>password</i> -a -f v3.user.ibm_oc.ldif

3. Repeat this procedure for each directory server in your integration.

---

**Important:** If you are working with an *existing* NetPoint installation, you need to manually extend the VDE schema using the step below.

---

4. Manually extend the VDE schema with NetPoint attributes using the VDE\_user\_schema\_add.ldif file below.

```
DN_ConversionTool_install_dir/oblix/tools/DataAnyWhere/OblixUserSchema/
VDE_user_schema_add.ldif
```

```
ldapmodify -h host -p port -D bind-dn -w password -a -f
```

5. Extend your VDE schema to represent all your back-end data sources as indicated below:

- **Either** add attributes from your back-end directory to the VDE schema:

Active Directory Example—Update/add attributes from “user” or “inetOrgPerson” object class to VDE inetOrgPerson object class.

Database Example—Add attributes from your Oracle Employees table to the inetOrgPerson object class.

- **Or** create a new object class having all visible attributes from the native data store.

Active Directory Example—Create a new object class (*MyCompanyPerson*) having all needed attributes from “user” or “inetOrgPerson” object class.

Database Example—Create a new object class (*MyCompanyPerson*) having all visible attributes from your Oracle Employees table to the inetOrgPerson object class.

---

**Note:** The schema extension can be done using the DME user interface (DME > *Your\_Project* > *Your\_Server* > Engine > Schema. When your extended schema is in an ldif file, use ldapmodify to load it into your VDE instance.

---

## Creating Mapping Files for Adapters

You are ready to create the mapping files needed for the data store adapters you will develop later:

- Each mapping file results in a filter that converts a back-end schema to the front-end (VDE) schema.
- Each mapping file enables you to map inbound and outbound data from the data store to remove anything inappropriate.

You can create your own mapping files from scratch or you can use the Oblix-provided sample files, which include several plug-ins. See “NetPoint-VDE Integration Templates” on page 708.

The steps below use the Oblix-provided samples, and are provided as a guide. The procedure is similar for both LDAP and RDBMS mapping files.

For details about using the DME, see your VDE documentation.

## To create a mapping file for a data store adapter

1. Complete activities in “Obtaining/Updating Sample Adapter and Mapping Templates” on page 659 so you have the sample mapping files provided by Oblix.
2. In the DME Server Navigator window, select your VDE server.
3. Select Engine > Mapping.
4. Right-click Mapping, then select New Mapping.

In the New Mapping window that opens, a File list contains the names of the sample mapping templates you copied to the DME earlier.

5. Specify the information requested:
  - **Server**—Specify the server containing your project.
  - **File template**—Choose the sample mapping template you want.
  - **File Name**—Enter a unique name for the version you will modify

You need to assign a new name so that your changes do not over write the sample template, which you should preserve for future use.

6. In the Name field, enter a unique name for the version you will modify.
7. Click Finish.

The name appears in the window on the left.

8. In the DME Server Navigator window, select the name of the file you just created to display it in the window on the right.

*Before* you finish NetPoint installation and setup you must customize your mapping file and add it to the data store adapter. You can customize the mapping file now, or later.

9. Continue as follows:
  - Deploy your mapping file to the server, as usual.
  - Proceed to “Creating Data Store Adapters” on page 669.

You can modify the mapping script for your needs now or after you create the data store adapter. See “Customizing Adapters and Mapping Files” on page 679. When you customize the file and include it file in an adapter:

- If you are *not* using the Oblix-provided sample file, you may need to create a dummy user (see “Unexpected Group Deletion Problem” on page 723.)
- If you are using the Oblix-provided sample, this occurs automatically.

# Creating Data Store Adapters

You now need to create an adapter for each data store you want to connect:

- **For Directories**—Create an LDAP adapter as discussed in “Creating Adapters for LDAP Directories” on page 669.
- **For Databases**—Create a database adapter as described in “Configuring a Database Adapter” on page 673.
- **For Split Profiles**—Create an individual adapter for each data store, then create an adapter to join the two data sources into a single view. See “Creating a Split-Profile Adapter” on page 675.
- **For Multiple Directories**—Create a separate adapter to connect each data source to the VDE adapter, as described in “Creating a Multiple-Directories Adapter” on page 677.

You can create an adapter from scratch or use the sample templates provided by Oblix, which provide you with a quick start. When you use a sample adapter template provided by Oblix, you need to fill in connection and credential information, logical root, remote root, and so on, to create the adapter. You also need to modify and/or tailor the settings for each data store. Once the adapter is created, the information defined in the template will be set for this adapter.

For details about Oblix templates, see “NetPoint-VDE Integration Templates” on page 708. For details about modifying templates, see “Customizing Adapters and Mapping Files” on page 679.

## Creating Adapters for LDAP Directories

The procedure for creating an LDAP adapter is similar regardless of the host directory server. You first create the LDAP adapter, then add plug-ins such as your mapping file.

As described below, ADAM and Active Directory adapters do include some requirements that other adapters do not.

**About Active Directory and ADAM Adapters**—Active Directory and ADAM require two adapters each: one for SSL that must be created first and a second for an open connection that must be created second. Oblix provides individual sample templates for each of these. Setting up this environment involves:

- Creating an Active Directory or ADAM adapter for an SSL connection
- Creating an Active Directory or ADAM adapter for an open connection

There are two plug-ins that Active Directory and ADAM adapters require. If you use the Oblix-provided sample templates, the following two plug-ins are *already*

*included*. However, if you create your own templates, you must add the following two plug-ins manually.

- **Active Directory Password Plug-In**—Active Directory and ADAM require the use of the secure mode to set or change a password.

To address performance concerns, a Password Only SSL mode is supported so that while normal operations are going through the adapter with the Open connection, operations related to password change/set functions are redirected to the adapter with the SSL connection.

---

**Note:** If you do *not* use the Oblix-provided sample templates, you need to add the SSL adapter you create as the Active Directory Password plug-in to the open-connection adapter you create, as described in “To create an adapter for LDAP” on page 670.

---

- **Active Directory Ranged Attributes Plug-In**—Active Directory and ADAM require the use of the Active Directory Ranged Attributes plug-in to handle the group page issue.

This plug-in concatenates all the group pages returned by Active Directory/ADAM and returns the information to the VDE client as one result.

---

**Note:** If you use the Oblix-provided sample templates, you simply edit the value. If you do *not* use the Oblix-provided sample templates, you must create and add the Active Directory Ranged Attributes plug-in to the open-connection adapter you create.

---

One generic procedure is given for all LDAP directories. This example includes steps to create two adapters for ADAM (assuming that you are not using an Oblix-provided example).

---

**Important:** When you use Oblix-provided templates for Active Directory or ADAM, see step 17 for details about the open connector you need to create. The SSL connector is included in the Oblix template.

---

### **To create an adapter for LDAP**

1. Complete activities in “Obtaining/Updating Sample Adapter and Mapping Templates” on page 659 so you have the sample adapter templates provided by Oblix.
2. In DME, navigate to Adapters > New > LDAP Adapter to display the Adapter configuration screen.
3. Select the appropriate adapter type from the Adapter Template list.

For example:

**Adapter Template:** OblixADAMSSLAdapterUsingMapper

4. Enter a unique adapter name.

For example:

**Adapter Name:** *CustomAdamSSLAdapter*

5. Fill in the server address, server proxy port, and server proxy bind DN for the LDAP Server to which you wish to connect.
6. Supply a Proxy Password and Passthrough credentials.

For example:

**Proxy Password:** *xxxxxxxx*

**Passthrough credentials:** Always

Specifying “Always” may impact performance; however, using “Bind Only” or “Never” is less secure.

Next you specify the connections options.

---

**Note:** When you do *not* use Oblix-provided templates or ADAM or Active Directory, you create an SSL adapter to include as a plug-in to an open connection adapter.

---

7. For Connection Options in an SSL version (not needed when using Oblix-provided templates), select:

**Connection Options:** Secure SSL/TLS

This step connects to the data store and downloads the certificate automatically.

8. For Remote Base, click the button labeled with an ellipsis (...).

A screen appears showing the searchbase (root DN) of the LDAP directory server you connected to.

At this point, you need to map the physical namespace to the logical namespace.

9. Select the remote physical namespace (searchbase) from the back-end data store.

For example:

*ou=company,c=us,dc=intranet,dc=pspl,dc=co,dc=in*

10. In the Mapped Namespace field, enter the logical VDE namespace.

For example, if your VDE root suffix is *o=MyCompany,c=us*, you can have a mapped namespace of:

*ou=ADAM,o=MyCompany,c=us*

**11.** Click Finish.

You should see the newly created LDAP adapter in the Server Navigator window, under the Adapter list.

**12.** Click the new Adapter name in the Server Navigator window:

- a) Click the Routing tab in the right pane.
- b) In General Settings, ensure that visibility is set to internal if this is an SSL adapter for Active Directory or ADAM.

For example:

General Settings

**Visibility:** Internal

- c) Click Finish.

For NetPoint to function properly, DN attributes used in NetPoint (like manger, secretary, uniqueMembers, and so on) need to be converted to the logical view format when viewed then back to the physical format when stored.

**13. Optional—DN Attributes—**

- a) Double click the adapter you created.
- b) In the right window, click on the “Config” tab.
- c) Under Settings, specify DN Attributes in a comma separated list of VDE attributes for all object classes/tables.

**14.** Right-click the adapter name in the Server Navigator window, then select Save to Server.

You have completed your first adapter and need to repeat this procedure for each data store in your integration.

When, you need to repeat this procedure to create a second adapter for ADAM, this time with an open connection.

---

**Note:** In the step below, only the *differences* between the SSL adapter you created above and the open connection adapter you need to create for ADAM and Active Directory are identified. All other specifications remain the same.

---

- 15. ADAM/Active Directory Open Connection Adapters**—Create the required open connection adapter by repeating the procedure above with the following *differences*.

For example:

**Adapter Template:** OblixADAMAdapterUsingMapper

**Adapter Name:** *CustomAdamOpenAdapter*

**Port:** *open\_port*

**Connection Options:** (Neither box should be checked) **Searchbase:** *same as the SSL adapter*

**Visibility:** Yes

- 16. Optional—DN Attributes**—For Active Directory or ADAM adapters created without the Oblix supplied sample template, this should be the same list as used in the SSL adapter created earlier.

- a) Double click the Active Directory adapter you created.
- b) In the right window, click on the “Config” tab.
- c) Under Settings, specify DN Attributes in a comma separated list of VDE attributes for all object classes/tables.

- 17.** Save and Deploy, as usual.

- 18.** Continue as follows:

- See “Editing an Adapter Plug-in to Refer to Your Mapping File” on page 696.
- See also “Customizing Adapters and Mapping Files” on page 679.
- Create other LDAP adapters as needed, or create any database adapters as described next.

## Configuring a Database Adapter

You can skip this procedure if it is not relevant to your environment.

The procedure below is a generic example. Your environment will vary.

### To configure a database adapter

1. Complete activities in “Deploying JDBC Driver Libraries for Your RDBMS” on page 660.
2. In DME, navigate to Adapters > New > Database Adapter to display the Adapter configuration screen.
3. Select the OblixDBAdapterUsingScript identified in “Database Template: OblixDBAdapterUsingScript” on page 716.
4. Enter a unique Adapter Name.

5. Enter logical namespace DN for the mapping.
6. Select “use predefined database”.
7. Select the Type of the database you plan to connect, such as MS SQL Server.
8. Fill in the host, port, database name, username and password for the database server.
9. Click Validate Connection to see if the connection information is correct, then click Next.
10. Proceed as follows for your environment:
  - **Other Templates**—When you are *not* using Oblix-provided templates, complete steps 11, 12, 13, and 14 as indicated below.
  - **Oblix-Provided Templates**—When using the Oblix-provided “Database Template: OblixDBAdapterUsingScript” on page 716, during steps 11, 12, 13, and 14 below you need *only* click Next.
11. On the database adapter mapping Choose table screen, complete the steps below:
  - a) Select the table you want to use from the left pane
  - b) Click “>” to move it to right pane.
  - c) Click Next.
12. On database adapter mapping: Build Joins screen, click Next to skip it.
13. On the database adapter mapping: map attributes screen, complete the steps below:
  - a) Click the logical DN you specified before.
  - b) Click Add (below the object class pane) to add the hierarchy.
  - c) In the pop up window, complete the following activities:
    - In object class, fill in the LDAP object class (such as inetorgperson) to which you want to map.
    - In the RDN field, fill in the RDN attribute name (such as cn).
    - Click Ok.
14. On database adapter mapping: map attributes screen, complete the steps below:
  - a) Click on the node you just created (for example, “cn= inetorgperson”).
  - b) Click Add (below the Attribute pane).
  - c) In the pop up window, select ldap attribute, table name, and table column.  
You can type in the LDAP attribute name (such as obuseraccountcontrol) if it is not on the list.

d) Continue this until all the attributes that you want to map are mapped.

---

**Note:** You need to map at least the `cn`, `uid`, `password`, and `obuseraccountcontrol` fields for COREid to work properly. Be sure that `obuseraccountcontrol` is Activated.

---

e) Add `password` and `obuseraccountcontrol` columns to an existing table in the database if table does not contain those columns.

15. Click Finish.

You should now see the newly created DB adapter in the Server Navigator window, under the Adapter list.

16. Right-click the Adapter name in the Server Navigator window, select Save to Server.

17. Check the Client view in the Browser pane to verify the configuration.

18. **All**—Continue as follows:

- Add a mapping file to this adapter, as described in “Editing an Adapter Plug-in to Refer to Your Mapping File” on page 696.
- See also “Customizing Adapters and Mapping Files” on page 679.
- Create other adapters as described next.

## Creating a Split-Profile Adapter

A split profile is one where you have the same users with different attributes stored on different directory servers.

The primary data store contains essential attributes while secondary data stores provide optional attributes. For example, suppose you have two different directory server types and an RDBMS. In this case, you need to create a split-profile adapter to join the views together.

Before you can create a split-profile adapter, you need to have the individual data store adapters created. Each primary *and* secondary adapter must have “Visibility” set to “Internal” so that only VDE will see them.

While you create the split-profile adapter, you identify the primary adapter and bind to that primary adapter. After creating the split-profile adapter, you specify join rules to identify the primary adapter and the first secondary adapter to be joined. While you specify join rules, you can indicate that you want to join the primary adapters to many secondary adapters (one to many).

---

**Note:** Oblix does *not* provide a Join View adapter template; however, OctetString does.

---

The searchbase (VDE refers to this as the root base) for a split-profile should be the same as that of the primary directory.

You need to ensure that the logical view of the split-profile adapter is the same as the primary data store. The split profile adapter does *not* map the values of DN attributes from Primary logical view to the split-profile logical view and vice versa.

The procedure below provides a general guide using the Join View method. For more information, see your VDE documentation. The adapters in this example include ADAM as the primary, and Sun as the secondary. In this case, you use the open-connection adapter you created for ADAM because it includes the appropriate plug-ins, including the SSL adapter. Your environment will vary.

### **To create a split-profile adapter**

1. Create an adapter for each data store you plan to join, as described in “Creating Adapters for LDAP Directories” on page 669.
2. In the DME Server Navigator window, select a VDE server > Engine.
3. Right-click Adapters, then select New > Join View Adapter.
4. In the dialog box, Adapter Template, select a default Join View template.
5. In the Adapter Name field, enter a unique name for your customized template.

For example:

**Adapter Name:** *CustomJoinADAMSun*

6. In the Adapter Suffix/Namespace list, enter the same namespace (base DN) with the DN of the primary adapter.

In this example, ADAM is the primary adapter. You must specify the name of the open-connection adapter, which includes the SSL adapter as a plug-in.

7. In Primary Adapter field, select your primary adapter.

For example:

**Primary Adapter:** *CustomAdamAdapter*

8. In the Binding Adapter list, select the *same* adapter.

For example:

**Binding Adapter:** *CustomAdamAdapter*

9. Click Finish.

The adapter name appears in the left pane and the Join View Primary Adapter Configuration window appears on the right.

10. In the Join View Primary Adapter Configuration window, Settings area, enter settings for the primary and binding adapters.

For example:

### Settings

**Primary Adapter:** *CustomAdamAdapter*

**Binding Adapter:** *CustomAdamAdapter*

11. Beside Join Rules, click the New button to display the Enter Join Rules dialog.
12. In the Enter Join Rules dialog, select the secondary adapter to join, then select a type class, and conditions for your environment.

For example:

**Joined Adapter:** *CustomSunAdapter*

**Type Class:** ... One to Many Joiner

**Conditions:** *cn=cn*

13. Repeat step 12 to join another adapter; otherwise, skip this step.
14. Right-click the adapter name in the Server Navigator window, select Save to Server.
15. Confirm the new configuration in the Browser window, Client View.

## Creating a Multiple-Directories Adapter

When you have multiple directory servers behind VDE, you need to create a local data store adapter entry within VDE, then add an entry for the VDE virtual root that is used as the searchbase for NetPoint, as outlined in below.

### Task overview: Creating a multiple-directories adapter

1. Create an adapter for each data store you plan to include, as described in:
  - “Creating Adapters for LDAP Directories” on page 669
  - “Configuring a Database Adapter” on page 673
2. Ensure that each directory server uses the *same* searchbase so the multiple-directories adapter will be the root.
3. Complete activities in “Creating an Local Data Store Adapter” on page 678.
4. Complete activities in “Creating a Physical Node for the Virtual Root” on page 679.

## Creating an Local Data Store Adapter

The only time a local store adapter is needed for integration with NetPoint is to create a virtual entry that is the parent of entries in multiple adapters so that it can appear as if a single contiguous tree exists.

For example, suppose you have two directories and want to create a directory tree with them:

**Directory 1**—*ou=Marketing,o=Company*

**Directory 2**—*ou=Product,o=Company*

In this case, to search from the *o=Company* level and have a search that covers both Directory 1 and Directory 2 you can use the local store adapter and create a single entry *o=Company* as its only entry. You would then have a full tree that looks like the one below:

```
o=Company—NetPoint can now search from here
/\
ou=Marketing ou=Product
```

The local data store adapter is needed only when:

- You want a unified searchbase for all individual data store adapters
- All individual data store adapters have the same root searchbase
- No duplicate entries exist in any data store (either remove the duplicate entries or filter them out)

The steps below are general. For more information, see your VDE documentation.

### To create an adapter entry in VDE

1. In DME, navigate to Adapters.
2. Right-click Adapters.
3. Select New > Local Store Adapter to display the Adapter configuration screen.
4. Provide the adapter suffix (the common virtual root base for all the adapters).
5. Save to the server, as usual.
6. Proceed to “Creating a Physical Node for the Virtual Root” on page 679.

## Creating a Physical Node for the Virtual Root

After creating a local data store adapter entry for multiple directories, you need to create a physical node in the VDE directory because NetPoint setup reads the configured node as the global searchbase. You create a physical node for the virtual root using the `ldp` utility, as usual.

For details about using the DME, see your VDE documentation.

### To create a physical node for the virtual root

1. Locate the `ldp` or `ldapmodify` utility.
2. Add an entry for the VDE virtual root.

For example, if your virtual root is `o=Company, c=us`, you add the entry:

```
dn:o=Company,c=us
```

```
Objectclass: organization
```

```
o: Company
```

3. Ensure that each directory server uses the same searchbase so the multiple-directories adapter will be the root.
4. Proceed to “Customizing Adapters and Mapping Files” on page 679.

## Customizing Adapters and Mapping Files

The following discussions provide specifics related to the integration with NetPoint:

- “Customization Examples” on page 680
- “Customizing General Settings for NetPoint” on page 694
- “Customizing Routing Settings” on page 696
- “Editing an Adapter Plug-in to Refer to Your Mapping File” on page 696

## Customization Examples

As mentioned earlier, you can create your own templates from scratch or customize the samples provided by Oblix. The two types of samples that Oblix provides are:

**Sample Adapter Templates**—Sample templates for vendor-specific adapter files that you can use as the basis for individual adapters that connect native data stores to VDE.

---

**Note:** Oblix-provided sample templates are specific to both a single data store and also to a specific *user-defined* schema. Your environment will vary.

---

**Sample Mapping Files**—Sample mapping files that you can use so that VDE can transform the schema (or database fields) used by native data stores to the logical schema used by the aggregated virtual directory made visible to NetPoint.

The following examples illustrate the type of modifications to Oblix-provided samples that you may want to make for your environment. The information contained in the examples, and the specific modifications made, are for illustration only. Your environment will vary.

- “Customized Mapping Script for Active Directory” on page 680
- “Customized Mapping Script for Oracle” on page 687
- “Customized Adapter for Oracle” on page 689

### Customized Mapping Script for Active Directory

Figure 42 shows a *customized* version of the Active Directory directory server *mapping file*. The starting point for this example is the Oblix-provided sample template for the Active Directory directory server and a specific *user-defined* schema that is not included here.

---

**Important:** The DNConversion Toolkit is a stand-alone package that you *must* download from the Oblix Web site to ensure that you have the latest version. Do *not* use the toolkit in the *COREid\_install\_dir*.

---

#### To see the types of mapping script changes made for Active Directory

1. In your DME console, create a mapping file using the Oblix sample OblixADMMapping file as a base (see “Creating Mapping Files for Adapters” on page 667).

```
DNConversionTool_install_dir\oblix\tools\DataAnyWhere  
\mapping_templates\OblixADMMapping
```

2. Modify your mapping file for your environment and compare your customized version to the one for Active Directory shown in Figure 42.
3. Save and deploy your mapping script, as usual.
4. Save your mapping script as a template for future use:
  - Right click the new mapping template name, for example *MyADMMapping*.
  - Select Save as template.

**Figure 42** Sample OblixADMMapping Script Template Customized for Active Directory

---

```

<?xml version="1.0" encoding="UTF-8" ?> ...
# Mapping template for: Custom Data sets
#
# Target DS: AD --- using static Auxiliary objectclass
# Target user objectclasses: User and group
# Target custom schema:
#   AD_custom_schema_add.ldif
#   AD.NET_custom_schema_add.ldif
#
# Functions:
# a. maps AD user to inetOrgPerson
# b. maps AD group to groupofuniqueNames
# c. filters out auxiliary class from objectclass in add/modify
# d. filter out AD system attributes
# e. set native flag useraccountcontrol when user is activated/deactivated
# f. set grouptype to 8
#
def inbound():
  #first rename the attributes
  renameAttribute({'uniqueMember':'member','owner':'managedby','uid':'samaccountname'})

  renameAttribute({'carlicense':'gencarlicense','departmentnumber':'gendepartmentnumber'})

  #temporary.
  removeAttribute('nsaccountlock')

  #map object class names
  revalueAttribute('objectclass','groupofUniqueNames','group')
  revalueAttribute('objectClass','inetOrgPerson','user')

  #If static auxiliary class is used on AD, AD does not like to mention
  #the auxiliary classes in the objectclass attribute. If dynamic auxiliary
  #class is used on AD, comment these out.
  removeAttributeValue('objectclass','person')

```

```

removeAttributeValue('objectclass','organizationalPerson')
#removeAttributeValue('objectclass','inetOrgPerson')
removeAttributeValue('objectclass','oblixOrgPerson')
removeAttributeValue('objectclass','oblixpersonpwdpolicy')
removeAttributeValue('objectclass','oblixadvancedgroup')
removeAttributeValue('objectclass','oblixgroup')
removeAttributeValue('objectclass','oblixAuxLocation')
#--- Remove custom data auxiliary object classes
removeAttributeValue('objectclass','genAuxLocation')
removeAttributeValue('objectclass','genAuxUserEquipment')
removeAttributeValue('objectclass','genAuxUserNetwork')
removeAttributeValue('objectclass','genAuxUserPersonal')
removeAttributeValue('objectclass','genAuxUserSecurity')

#If static auxiliary class is used in AD, remove the objectclass attribute
#during modify. AD does not like the mentioning of the auxiliary class.
if operation == 'modify':
    removeAttribute('objectClass')

#set the native flag useraccountcontrol based on the value of obuseraccountcontrol.
if haveAttribute('obuseraccountcontrol'):
    copyAttribute('obuseraccountcontrol','userAccountControl')
    #during modify, read the user entry first.
    if operation == 'modify':
        currentUser = getByname(name)
        val = int(`getAttributeValues(currentUser,'userAccountControl')[0]`)
    else:
        val = 546
    #Deactivate - set the 2nd bit
    revalueAttribute('userAccountControl','Obwfpendingactivate`,`val | 0x0002`)
    revalueAttribute('userAccountControl','DEACTIVATED`,`val | 0x0002`)
    revalueAttribute('userAccountControl','Obwfpendingdeactivate`,`val | 0x0002`)
    #Activate - set the 2nd bit
    revalueAttribute('userAccountControl','ACTIVATED`,`val & ~0x0002`)

#when adding a group entry, add the grouptype and samaccountname.
#groupType is hard coded here. If multiple group types are to be supported,
#configured grouptype in VDE for user to enter.
if operation == 'add':
    if haveAttributeValue('objectClass','group'):
        addAttributeValue('groupType','8')
        if not haveAttribute('samaccountname'):
            copyAttribute('cn','samaccountname')
        #remove these attributes as they are not in AD group. It is better not
        #configure them in COREid if not used.

```

```

    #removeAttribute ('businessCategory')
    removeAttribute ('seeAlso')
    removeAttribute ('o')

    #if haveAttributeValue('objectClass','user'):
        #removeAttributeValue('objectClass','person')
        #removeAttributeValue('objectClass','organizationalPerson')
        #removeAttributeValue('objectClass','inetOrgPerson')

if operation == 'modify':
    currentEntry = getByname(name)
    val = getAttributeValues(currentEntry,'objectClass')
    if DirectoryString('group') in val:
        #removeAttribute ('businessCategory')
        removeAttribute ('seeAlso')
        removeAttribute ('o')

#filter out obgroupcreator otherwise ipplanet user can not create ad group.
if haveAttribute ('obgroupcreator'):
    removeAttribute ('obgroupcreator')

return

def outbound():
    #first rename the attributes
    renameAttribute({'member':'uniqueMember','managedby':'owner','samaccountname':'uid'})
    renameAttribute({'gencarlicense':'carlicense','gendepartmentnumber':'departmentnumber'})

    #map object class names
    revalueAttribute('objectClass','group','groupofUniqueNames')
    revalueAttribute('objectClass','user','inetOrgPerson')

    #filter out AD system attributes
    removeAttribute ('allowedAttributes')
    removeAttribute ('allowedAttributesEffective')
    removeAttribute ('allowedChildClasses')
    removeAttribute ('allowedChildClassesEffective')
    removeAttribute ('assistant')
    removeAttribute ('bridgeheadServerListBL')
    removeAttribute ('canonicalName')
    removeAttribute ('createTimeStamp')
    removeAttribute ('department')
    removeAttribute ('distinguishedName')
    removeAttribute ('dsASignature')

```

```
removeAttribute ('dScorePropagationData')
removeAttribute ('extensionName')
removeAttribute ('flags')
removeAttribute ('fromEntry')
removeAttribute ('frsComputerReferenceBL')
removeAttribute ('FRSMemberReferenceBL')
removeAttribute ('FSMORoleOwner')
removeAttribute ('generationQualifier')
removeAttribute ('instanceTyp')
removeAttribute ('isCriticalSystemObject')
removeAttribute ('isDeleted')
removeAttribute ('isPrivilegeHolder')
removeAttribute ('lastKnownParent')
removeAttribute ('managedObjects')
removeAttribute ('modifyTimeStamp')
removeAttribute ('mS-DS-ConsistencyChildCount')
removeAttribute ('mS-DS-ConsistencyGuid')
removeAttribute ('name')
removeAttribute ('netbootSCPBL')
removeAttribute ('nonSecurityMemberBL')
removeAttribute ('nTSecurityDescriptor')
removeAttribute ('objectCategory')
removeAttribute ('objectGUID')
removeAttribute ('objectVersion')
removeAttribute ('partialAttributeDeletionList')
removeAttribute ('partialAttributeSet')
removeAttribute ('possibleInferiors')
removeAttribute ('queryPolicyBL')
removeAttribute ('replPropertyMetaData')
removeAttribute ('replUpToDateVector')
removeAttribute ('revision')
removeAttribute ('SDRightsEffective')
removeAttribute ('serverReferenceBL')
removeAttribute ('showInAdvancedViewOnly')
removeAttribute ('siteObjectBL')
removeAttribute ('subRefs')
removeAttribute ('subSchemaSubEntry')
removeAttribute ('systemFlags')
removeAttribute ('uSNChanged')
removeAttribute ('uSNCreated')
removeAttribute ('uSNSALastObjRemoved')
removeAttribute ('USNIntersite')
removeAttribute ('uSNLastObjRem')
removeAttribute ('uSNSource')
removeAttribute ('wbemPath')
```

```
removeAttribute ('wellknownObjects')
removeAttribute ('whenChanged')
removeAttribute ('whenCreated')
removeAttribute ('instanceType')
removeAttribute ('ms-sql-olapcube')
removeAttribute ('ms-sql-database')
removeAttribute ('ms-sql-server')
removeAttribute ('ms-sql-sqlpublication')
removeAttribute ('ms-sql-sqldatabase')
removeAttribute ('ms-sql-sqlrepository')
removeAttribute ('ms-sql-sqlserver')
removeAttribute ('acpolarity')
removeAttribute ('acsubnet')
removeAttribute ('msexchconfigurationcontainer')
removeAttribute ('msmqconfiguration')
removeAttribute ('msmqenterprisesettings')
removeAttribute ('msmqmigrateduser')
removeAttribute ('msmqqueue')
removeAttribute ('msmqsettings')
removeAttribute ('msmqsitelink')
removeAttribute ('ntdsconnection')
removeAttribute ('ntdsdsa')
removeAttribute ('ntdsservice')
removeAttribute ('ntdssitesettings')
removeAttribute ('ntfrsmember')
removeAttribute ('ntfrsreplicaset')
removeAttribute ('ntfrssettings')
removeAttribute ('ntfrssubscriber')
removeAttribute ('ntfrssubscriptions')
removeAttribute ('accountExpires')
removeAttribute ('acSPolicyName')
removeAttribute ('adminCount')
removeAttribute ('badPasswordTime')
removeAttribute ('badPwdCount')
removeAttribute ('codePage')
removeAttribute ('controlAccessRights')
removeAttribute ('dBCSPwd')
removeAttribute ('defaultClassStore')
removeAttribute ('desktopProfile')
removeAttribute ('dynamicLDAPServer')
removeAttribute ('groupMembershipSAM')
removeAttribute ('groupPriority')
removeAttribute ('groupsToIgnore')
removeAttribute ('homeDirectory')
removeAttribute ('homeDrive')
```

```
removeAttribute ('lastLogoff')
removeAttribute ('lastLogon')
removeAttribute ('lmPwdHistory')
removeAttribute ('localeID')
removeAttribute ('lockoutTime')
removeAttribute ('logonCount')
removeAttribute ('logonHours')
removeAttribute ('logonWorkstation')
removeAttribute ('mSMQDigests')
removeAttribute ('mSMQDigestsMig')
removeAttribute ('mSMQSignCertificates')
removeAttribute ('mSMQSignCertificatesMig')
removeAttribute ('msNPAllowDialin')
removeAttribute ('msNPCallingStationID')
removeAttribute ('msNPSavedCallingStationID')
removeAttribute ('msRADIUSCallbackNumber')
removeAttribute ('msRADIUSFramedIPAddress')
removeAttribute ('msRADIUSFramedRoute')
removeAttribute ('msRADIUSServiceType')
removeAttribute ('msRASSavedCallbackNumber')
removeAttribute ('msRASSavedFramedIPAddress')
removeAttribute ('msRASSavedFramedRoute')
removeAttribute ('networkAddress')
removeAttribute ('ntPwdHistory')
removeAttribute ('operatorCount')
removeAttribute ('otherLoginWorkstations')
removeAttribute ('preferredOU')
removeAttribute ('primaryGroupID')
removeAttribute ('profilePath')
removeAttribute ('pwdLastSet')
removeAttribute ('scriptPath')
removeAttribute ('servicePrincipalName')
removeAttribute ('userAccountControl')
removeAttribute ('userParameters')
removeAttribute ('userSharedFolder')
removeAttribute ('userSharedFolderOther')
removeAttribute ('userSMIMECertificate')
removeAttribute ('userWorkstations')
removeAttribute ('masteredBy')
removeAttribute ('maxStorage')
removeAttribute ('userPrincipalName')
removeAttribute ('objectSid')
removeAttribute ('samAccountType')
removeAttribute ('badPasswordCount')
removeAttribute ('SAMAccountControl')
```

```
removeAttribute ('AdsPath')
removeAttribute ('directReport')

return
</ldap>
</adapters>
```

---

## Customized Mapping Script for Oracle

Figure 43 shows the sample `OblixDBMapping` file as it looks *after* being customized for an Oracle database that uses the SQL server as a back end with a *user-defined* schema. The original sample is located in:

```
DNConversionTool_install_dir\oblix\tools\DataAnywhere\mapping_templates
\OblixDBMapping
```

---

**Important:** The DNConversion Toolkit is a stand-alone package that you *must* download from the Oblix Web site to ensure that you have the latest version. Do *not* use the toolkit in the `COREid_install_dir`.

---

You can compare the original Oblix-provided sample with the one in Figure 43, to see the types of changes that are needed.

### To see mapping script changes for the Oracle database

1. In your DME console, create a mapping file using the Oblix sample `OblixDBMapping` file as a base (see “Creating Mapping Files for Adapters” on page 667).
2. Modify the mapping file for your environment and include the work around shown in Figure 43 (see also “Unexpected Group Deletion Problem” on page 723).
3. Save and deploy your mapping script, as usual.
4. Save your mapping script as a template for future use:
  - Right click the new mapping template name, for example `MyOracleDBMapping`.
  - Select Save as template.
5. See also the customized adapter for the Oracle database in Figure 44.

**Figure 43** Sample Mapping Customized for an Oracle Database

---

```
<?xml version="1.0" encoding="UTF-8"?>
<variables>
</variables>
<content>
def inbound():
    #These Oblix attributes are not being used. Remove them.
    removeAttribute('obver')
    removeAttribute('nsaccountlock')

    # More custom mapping
    # ....

    # If your user password is stored as character type, for example
    # NVARCHAR, CHAR, VARCHAR, etc, you need to map userPassword attribute
    # from binary syntax.
    mapSyntax('userPassword', 'IA5String')

    # This is a workaround ... for more information, see "Unexpected Group Deletion
    Problem" on page 723.
    # Need to prevent COREid from writing dummy user to backend database
    if haveAttributeValue('uniqueMember', 'cn=Dummy User'):
        #removeAttributeValue('uniqueMember', 'cn=Dummy User')
        if operation != 'modify':
            removeAttributeValue('uniqueMember', 'cn=Dummy User')
        else:
            change = removeAttribute('uniqueMember')[0]
            change.values.remove(DistinguishedName('cn=Dummy User'))
            addEntryChange(change)

    #Filter out objectclass. Only mention the structure class during add.
    if operation == 'modify':
        removeAttribute('objectClass')
    if operation == 'add':
        newobj = ''
        if haveAttributeValue('objectClass', 'inetOrgPerson'):
            newobj = 'inetOrgPerson'
        if haveAttributeValue('objectClass', 'groupOfUniqueNames'):
```

```

    newobj = 'groupOfUniqueNames'
    removeAttribute ('businessCategory')
    removeAttribute ('seeAlso')
    removeAttribute ('o')
    if haveAttributeValue('objectClass','oblixlocation'):
        newobj = 'oblixlocation'
    if not newobj == '':
        removeAttribute('objectClass')
        addAttributeValue('objectClass',newobj)

return

def outbound():
    #code here for handling outbound mapping
    # .....
    # This is a workaround to bug #18865
    if operation=='entry':
        # Add the following workaround for each multiple value DN attribute
        if haveAttribute('uniqueMember') and len(findFilters('uniqueMember')) > 0:
            addAttributeValue('uniqueMember','cn=Dummy User')
    return
</content>

```

---

## Customized Adapter for Oracle

Figure 44 shows a sample adapter after being customized for the Oracle database. The Oblix-supplied sample template was used as a starting point.

**Figure 44** Sample Adapter After Customization for the Oracle Database

---

```

<?xml version="1.0" encoding="UTF-8"?>
<adapters dirty="" version="0"
  xmlns="http://www.octetstring.com/schemas/Adapters" xmlns:adapters="http://
www.w3.org/2001/XMLSchema-instance">
  <dataBase dirty="" id="DB Adapter Company Employees" version="0">
    <root>ou=Employees,o=MyCompanyDB,c=us</root>
    <active>true</active>
    <routing>
      <critical>true</critical>

```

```

<priority>50</priority>
<inclusionFilter/>
<exclusionFilter/>
<plugin/>
<retrieve/>
<store>
  <exclude>carlicense</exclude>
  <exclude>street</exclude>
  <exclude>employeeType</exclude>
</store>
<visible>Internal</visible>
<levels>-1</levels>
<bind>true</bind>
<bind-adapters/>
<views/>
<dnpattern/>
</routing>
<pluginChains xmlns="http://www.octetstring.com/schemas/Plugins">
  <plugins>
    <plugin>
      <name>MyOracleDBMapping</name>
      <class>com.octetstring.vde.chain.plugins.mapper.Mapper</class>
      <initParams>
        <param name="mapfile" value="MyOracleDBMapping.mpy"/>
      </initParams>
    </plugin>
    <plugin>
      <name>Dump after</name>

<class>com.octetstring.vde.chain.plugins.DumpTransactions.DumpTransactions</
class>
      <initParams>
        <param name="loglevel" value="info"/>
      </initParams>
    </plugin>
    <plugin>
      <name>Dump before</name>

<class>com.octetstring.vde.chain.plugins.DumpTransactions.DumpTransactions</
class>

```

```

    <initParams>
      <param name="loglevel" value="info"/>
    </initParams>
  </plugin>
</plugins>
<default>
  <plugin name="Dump before"/>
  <plugin name="MyOracleDBMapping"/>
  <plugin name="Dump after"/>
</default>
</pluginChains>
<driver>oracle.jdbc.driver.OracleDriver</driver>
<url>jdbc:oracle:thin:@127.0.0.1:1521:QA2</url>
<user>CUSTDATA</user>
<password>oblix</password>
<ignoreObjectClassOnModify>>false</ignoreObjectClassOnModify>
<includeInheritedObjectClasses>>true</includeInheritedObjectClasses>
<maxConnections>10</maxConnections>
<mapping>
  <joins/>
  <objectClass name="inetOrgPerson" rdn="cn">
    <attribute field="EMPLOYEE_ID" ldap="uid"
      table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
    <attribute field="NAME" ldap="cn" table="CUSTDATA.EMPLOYEES"
type="VARCHAR"/>
    <attribute field="FIRST_NAME" ldap="givenName"
      table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
    <attribute field="LAST_NAME" ldap="sn"
      table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
    <attribute field="TITLE" ldap="title" table="CUSTDATA.EMPLOYEES"
type="VARCHAR"/>
    <attribute field="USERPASSWORD" ldap="userPassword"
      table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
    <attribute field="PREFERREDLANGUAGE"
      ldap="PreferredLanguage" table="CUSTDATA.EMPLOYEES" type="CHAR"/>
    <attribute field="MAIL" ldap="mail" table="CUSTDATA.EMPLOYEES"
type="VARCHAR"/>
    <attribute field="CHALLENGEPHRASE" ldap="ChallengePhrase"
      table="CUSTDATA.EMPLOYEES" type="CHAR"/>
  </objectClass>
</mapping>

```

```

<attribute field="PHOTO" ldap="Photo"
  table="CUSTDATA.EMPLOYEES" type="BLOB"/>
<attribute field="DESCRIPTION" ldap="Description"
  table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
<attribute field="OBUSERACCOUNTCONTROL"
  ldap="OBUSERACCOUNTCONTROL" table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
<attribute field="OBLOGINTRYCOUNT" ldap="oblogintrycount"
  table="CUSTDATA.EMPLOYEES" type="NUMERIC"/>
<attribute field="OBPASSWORDCREATIONDATE"
  ldap="obpasswordcreationdate" table="CUSTDATA.EMPLOYEES" type="VARCHAR"/
>

<attribute field="OBPASSWORDHISTORY" ldap="obpasswordhistory"
  table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
<attribute field="OBPASSWORDCHANGEFLAG"
  ldap="obpasswordchangeflag" table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
<attribute field="OBPASSWORDEXPMAIL" ldap="obpasswordexpmail"
  table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
<attribute field="OBLOCKOUTTIME" ldap="oblockouttime"
  table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
<attribute field="OBFIRSTLOGIN" ldap="obfirstlogin"
  table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
<attribute field="OBRESPONSETRIES" ldap="obresponsetries"
  table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
<attribute field="OBLASTLOGINATTEMPTDATE"
  ldap="oblastloginattemptdate" table="CUSTDATA.EMPLOYEES" type="VARCHAR"/
>

<attribute field="OBLASTRESPONSEATTEMPTDATE"
  ldap="oblastresponseattemptdate" table="CUSTDATA.EMPLOYEES"
type="VARCHAR"/>
<attribute field="OBRESPONSETIMEOUT" ldap="obresponsetimeout"
  table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
<attribute field="MANAGER_DN" ldap="Manager"
  table="CUSTDATA.EMPLOYEES" type="VARCHAR"/>
</objectClass>
<objectClass name="groupOfUniqueNames" rdn="cn">
  <attribute field="GROUP_NAME" ldap="cn"
    table="CUSTDATA.GROUPS" type="VARCHAR"/>
  <attribute field="OWNER_DN" ldap="owner"
    table="CUSTDATA.GROUPS" type="VARCHAR"/>

```



```

        table="CUSTDATA.GROUPS" type="VARCHAR"/>
</objectClass>
<objectClass name="oblxllocation" rdn="obid">
  <attribute field="OBID" ldap="obid"
    table="CUSTDATA.OBLIXLOCATION" type="VARCHAR"/>
  <attribute field="OBLOCATIONNAME" ldap="oblocationname"
    table="CUSTDATA.OBLIXLOCATION" type="VARCHAR"/>
  <attribute field="OBLOCATIONTITLE" ldap="oblocationtitle"
    table="CUSTDATA.OBLIXLOCATION" type="VARCHAR"/>
  <attribute field="OBPARENTLOCATIONDN" ldap="obparentlocationdn"
    table="CUSTDATA.OBLIXLOCATION" type="VARCHAR"/>
  <attribute field="OBRECTANGLE" ldap="obrectangle"
    table="CUSTDATA.OBLIXLOCATION" type="VARCHAR"/>
  <attribute field="OBPHOTO" ldap="obphoto"
    table="CUSTDATA.OBLIXLOCATION" type="BLOB"/>
</objectClass>
</mapping>
</dataBase>
</adapters>
...

```

---

## Customizing General Settings for NetPoint

General adapter configuration and setup information is provided in the VDE/DME product manual. Once an adapter is created, default values are valid for most places. The following highlights are concerns with NetPoint:

- DN attributes
- Connection Information

### DN Attributes—

- DN attributes should be set with the attribute names that are in DN syntax. These DN attributes could exist in the customer schema, or could be introduced by NetPoint auxiliary classes. This tells VDE to store the values of these DNs in their native form instead of logical form.
- The DN attributes are related to the object classes you are using:
  - **For inetorgperson and groupofuniqueNames**, use  
uniquemember, manager, secretary, owner

- **For user and group**, use  
member, memberOf, managedObjects, distinguishedname, objectcategory, manager, secretary, managedby
- **For NetPoint introduced auxiliary classes**, use  
obgroupadministrator, obgroupcreator
- **In Pass-Through Mode**, select  
Always

**Connection Information**—Set the following connection information in VDE based on the estimated workload:

Operation timeout  
Max Pool Connection  
Max Pool Wait  
Max Pool Tries

### **To customize general settings for NetPoint**

1. Review the information above.
2. In the DME Server Navigator window, select a server, then locate and select the name of the adapter.
3. In the right pane, click the Routing tab.
4. In General Settings, ensure that visibility is set to internal.

For example:

General Settings

**Visibility:** Internal (*for split profile adapters*)

5. Ensure that your adapter uses the settings discussed in this section.
6. Click Finish.
7. Right-click the adapter name in the Server Navigator window, then select Save to Server.

## Customizing Routing Settings

If you are setting up an adapter that will be used by the Join View adapter, the visibility of the primary and secondary adapters should be set to “internal” so they are only invoked by the Join View adapter.

Once your adapters are working, you should observe the performance and evaluate the log to see if there is a pattern that an adapter is unnecessarily invoked by certain operation. If yes, you should try to use the following to filter block the unnecessary operation for that particular adapter. This step is very important to improve the overall performance:

- Filter to include
- Filter to exclude
- DN matching

### To customize routing settings

1. Select the Adapter name in the Server Navigator window.
2. Click the Routing tab in the right pane.
3. Select the options for your environment:
  - Filter to include
  - Filter to exclude
  - DN matching
4. Save to the server, as usual.

## Editing an Adapter Plug-in to Refer to Your Mapping File

The Oblix-provided sample adapter templates include several plug-ins hooked in already, as discussed in “NetPoint-VDE Integration Templates” on page 708. There are two types of plug-ins:

- **Plug-in**—A predefined plug-in that provides a parameter-based user interface for configuration.
- **Mapping Plug-in**—A plug-in for a mapping script.

Keep the following in mind as you work:

- If you use the Oblix-provided sample templates, you need only modify plug-ins.
- If you do *not* use the Oblix-provided sample templates you need to add plug-ins to your adapter. For example:

As discussed earlier, Active Directory and ADAM adapters require two specific plug-ins, which are included in the Oblix-provided sample templates.

If you do *not* use the Oblix-provided sample templates, you need to add the SSL adapter you create as the Active Directory Password plug-in to the open-connection adapter you create and also add the Active Directory Ranged Attributes plug-in. Be sure to specify the same mapped namespace as the SSL adapter. See “Creating Adapters for LDAP Directories” on page 669:

The following is an example only. For details about using DME, see your VDE documentation.

### To edit an adapter plug-in to refer to your mapping file

1. Complete activities in:
  - “Creating Mapping Files for Adapters” on page 667
  - “Creating Data Store Adapters” on page 669
2. In the DME Server Navigator window, select a project and server, then locate and select the name of the adapter to which you want to add or verify a plug-in.
3. In the right pane, click the Plug-ins tab.
4. On the Adapter Plug-ins screen:
  - a) Click “All Operations” to see which plug-ins are included.

For ADAM or Active directory, you need the following plug-ins in the order shown below:

Active Directory Ranged Attributes  
OblixADMMapping (*should be the mapping file you created earlier*)  
Active Directory Password
  - b) Arrange plug-ins for your environment using the up and down arrows.
  - c) When the mapping file you created earlier is *not* listed:
    - Select the current mapping, for example, *OblixADMMapping*, then click the Edit button.
    - In the Name field, change the name to your mapping file name (for example *MyADMMapping*).
    - In the Map File list, select you mapping file (for example *MyADMMapping.mpy*).
  - d) When using *non-Oblix templates* for Active Directory or ADAM adapters, add the SSL adapter you created earlier to handle the password.
    - Click the New Plug-in button.
    - Click Select from Server, then select the Active Directory Password plug-in.
    - Specify your SSL adapter name as the value of the parameter, for example: *CustomAdamSSLAdapter*.

- Select a parameter line, then click Edit.
  - Specify your ADAM or Active Directory SSL adapter name as the value, for example: *CustomAdamSSLAdapter*.
5. Finish, save, and deploy as usual.

## Completing NetPoint Installation and Setup

Now that you have completed *all* other essential activities, described above, you are ready to complete NetPoint installation and setup.

### To complete NetPoint installation and setup

1. Complete all tasks above.
2. **WebPass**—Install WebPass as described in the *NetPoint 7.0 Installation Guide*.
3. **COREid System Set Up**—Set up the COREid System using specifications below, then finish setup as you normally do:
  - a) **User Data Directory Server**—Select COREid Data Anywhere as the directory type.
  - b) **User Data ... Host**—Specify the machine hosting VDE.
  - c) **User Data ... Port**—Specify the VDE LDAP licensing port.
  - d) **Searchbase**—Specify the VDE virtual DN, which may be any part of the virtual tree.
  - e) **Oblix Data Directory Server**—Select the native directory you specified during COREid Server installation—NetPoint configuration (and policy data) must be stored outside the VDE virtual directory.
  - f) **Automatically Update Schema**—Select Yes to automatically update the VDE schema with NetPoint auxiliary attributes.
  - g) **Automatically Configure Person and Group Object Classes**—Choose Yes or No, as you normally do, to configure the VDE schema.

**Note:** For details about manually configuring Person and Group object classes, see the *NetPoint 7.0 Installation Guide*.
4. **Access Manager**—Install and set up the Access Manager as described in the *NetPoint 7.0 Installation Guide* using specific details indicated below, then complete setup as usual:
  - a) Specify *user* data directory server details during setup, as described above.
  - b) Specify the following during Access Manager setup:

- Searchbase—*Must* be the same searchbase you specified during COREid System setup.
  - Configuration DN—*Must* be the same NetPoint configuration data DN you specified during COREid System setup.
  - Policy Base—*Must* be the same policy data DN you specified during Access Manager installation.
5. **Access Server**—Install the Access Server as described in the *NetPoint 7.0 Installation Guide* and below:
    - a) Provide information for the Oblix configuration data directory server.
    - b) Identify where the NetPoint policy data is stored.
    - c) Provide the following information when asked:
      - Access Server ID
      - Configuration DN—*Must* be the same as specified earlier.
      - Policy Base—*Must* be the same as specified earlier.
  6. **WebGate**—Install the WebGate as described in the *NetPoint 7.0 Installation Guide*.
  7. **Failover**—Configure failover as described in:
    - “Failover Support” on page 648
    - The *NetPoint 7.0 Deployment Guide*
    - Your product documentation.

## Testing Your Integration

To test your integration, simply perform a COREid function that requires obtaining user data from a native directory.

If the operation works, the integration is a success.

# Reference Information

The following sections provide technical details related to several aspects of the NetPoint-VDE integration.

- “NetPoint Auxiliary Attributes” on page 700
- “About the Data Anywhere Toolkit” on page 703

---

**Note:** See also “About the DN Conversion Tool” on page 706.

---

- “NetPoint-VDE Integration Templates” on page 708

See also details about “NetPoint-VDE Integration Templates” on page 708.

## NetPoint Auxiliary Attributes

Certain NetPoint functions require that specific attributes exist in the schema of both your top-level virtual directory and in the schema (or database equivalent) of each target data store.

- You can extend your virtual directory schema automatically as follows:
  - When you install and set up the COREid System you have the opportunity to automatically (or manually) extend the VDE schema when you choose COREid Data Anywhere as the user data directory server. See the *NetPoint 7.0 Installation Guide* (part number NPINST7.0B).
  - After you upgrade an older NetPoint installation to v7.0.2 you manually extend the VDE schema using the ldapmodify utility as discussed in “Extending Directory Schemas” on page 665.
- You extend the target LDAP directory schemas by running the ldapmodify.exe utility with the appropriate ldif file, as described in “Extending Directory Schemas” on page 665.
- You simulate the object classes of your virtual directory by mapping all the user accounts in your primary database tables to the appropriate classes. See “About Adding Attributes to Target Database Tables” on page 638.
- You simulate the auxiliary NetPoint user attributes that enable special NetPoint features by creating extra columns in your primary database tables. See “About Adding Attributes to Target Database Tables” on page 638.

You use NVARCHAR with length of 1000 for all unbounded and/or user data, and VARCHAR with a length of 240 for all NetPoint-specific valued attributes.

Table 49 on page 701 correlates the NetPoint auxiliary attributes required for specific User Manager functions.

---

**Note:** The NetPoint-VDE integration does *not* support the Oblix Location object; therefore the User Location function is *not* supported for the User Manager application.

---

**Table 49** Extended Attributes Required by User Manager Functions

User Manager Function	Required Attributes	Suggestions for Attribute Type and Length
User Add/Activate/Deactivate	Obuseraccountcontrol	VARCHAR (240)
Workflow Surrogate	oboutofofficeindicator	VARCHAR (240)
Password Change on Reset	obpasswordchangeflag	VARCHAR (240)
Password Number of login tries allowed	oblogintrycount	VARCHAR (240)
Password Validity period	obpasswordcreationdate	VARCHAR (240)
Password Expiry Notice Period	obpasswordcreationdate	VARCHAR (240)
Password Lockout Duration	oblockouttime	VARCHAR (240)
Password Login tries reset	oblastloginattemptdate	VARCHAR (240)
Password minimum age	obpasswordcreationdate	VARCHAR (240)
Password history	obpasswordhistory password history support optional	NVARCHAR (1000)
Challenge Response	Customer attributes for Challenge phrase and response	NVARCHAR (1000)
Challenge Response Login tries reset	oblastresponseattemptdate	VARCHAR (240)
Challenge Response Lockout Duration	Obresponsetimeout oblockouttime	VARCHAR (240) VARCHAR (240)
Challenge Response Number of login tries allowed	obresponsetries	VARCHAR (240)

Table 50 correlates the NetPoint auxiliary attributes required for specific Group Manager functions.

**Table 50** Extended Attributes Required by Group Manager Functions

<b>Group Manager Function</b>	<b>Required Attributes</b>	<b>Suggestions for Attribute Type and Length</b>
Subscription type	obgroupsubscriptiontype	VARCHAR (240)
Group expansion	obgroupexpandeddynamic	VARCHAR (240)
Pure dynamic group	obgrouppuredynamic	VARCHAR (240)
Group administrators	obgroupadministrator The virtual directory must support exactly one administrator per group.	NPVARCHAR (1000)
Subscription message	obgroupsubscribemessage	NPVARCHAR (1000)
Unsubscription message	obgroupunsubscribemessage	NPVARCHAR (1000)
Subscription filters	obgroupsubscriptionfilter The virtual directory must support exactly one subscription per group.	NPVARCHAR (1000)
Subscription notification types	obgroupsubscribenotification Either subscription or unsubscription notification can be implemented, but both functions can not be implemented simultaneously.	NPVARCHAR (1000)
Dynamic filters	obgroupdynamicfilter The virtual directory must support exactly one dynamic filter per group	NPVARCHAR (1000)
Simplified access control	obgroupsimplifiedaccesscontrol	VARCHAR (240)
Group types	obgroupstype The virtual directory must support exactly one group type per group.	VARCHAR (240)
Selectable subscription types	obsubscriptiontypes The virtual directory must support exactly one subscription type per group. The possible subscription types are: Open, Close, Open with filter, and Controlled through workflow.	NPVARCHAR (1000)

## About the Data Anywhere Toolkit

The Data Anywhere toolkit is for use when you have an existing NetPoint installation and you want to integrate with VDE. It will convert all the user data related native DN suffixes in oblix/policy tree to logical DN suffixes.

The DNConversion Toolkit is a stand-alone package that you *must* download from the Oblix Web site to ensure that you have the latest version. Do *not* use the toolkit in the *COREid\_install\_dir*. The directory structure is shown in Figure 45.

*DNConversionTool\_install\_dir/oblix/tools/DataAnyWhere*

**Figure 45** Data Anywhere Toolkit Directory Structure

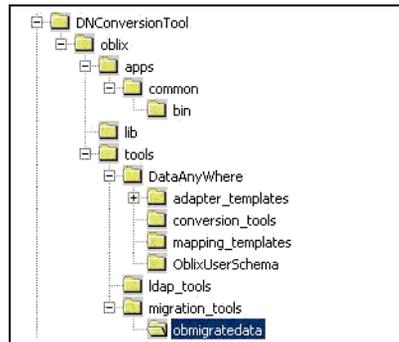


Table 51 identifies the contents of the toolkit.

**Table 51** Contents of the Data Anywhere Toolkit for Integrating NetPoint and VDE

Subdirectory	File Components	Description
n/a	README	An overview of the content, runtime requirements, and simple usage examples for the components of the Data Anywhere toolkit.
\OblixUserSchema	ADUserSchema.Idif ADAuxSchema.Idif ADAM_user_schema_add.Idif ADAMAuxSchema.Idif iPlanet_user_schema_add.Idif NDS_user_schema_add.Idif NDS_user_schema_delete.Idif v3.user.ibm_at.Idif v3.user.ibm_oc.Idif iPlanet_user_schema_delete.Idif  schema.oblix.xml  VDE_user_schema_add.Idif VDE_user_schema_delete.Idif  <b>Note:</b> When possible, use the Idif files provided with your COREid Server installation as these have not changed since NetPoint 7.0.	schema.oblix.xml extends the NetPoint user schema into your virtual directory  VDE_user_schema_add.Idif extends the VDE schema with NetPoint attributes.  The other files extend the NetPoint user schema to the directory servers supported by the NetPoint-VDE integration.
\adapter_templates \Database	OblixDBAdapterScriptUsingScript_adapter_template.XML	A basic template for creating a database adapter.
\adapter_templates \Ldap	<ul style="list-style-type: none"> <li>• OblixADAdapterUsingMapper_adapter_template.xml</li> <li>• OblixADAdapterUsingScript_adapter_template.xml</li> <li>• OblixADSSLAdapterUsingMapper_adapter_template.xml</li> <li>• OblixADAMAdapterUsingMapper_adapter_template.xml</li> <li>• OblixADAMAdapterUsingScript_adapter_template.xml</li> <li>• OblixADAMSSLAdapterUsingMapper_adapter_template.xml</li> </ul> For additional information, see “NetPoint-VDE Integration Templates” on page 708.	VDE adapter templates for directory servers that require templates. These can serve as a starting point for adapter creation. Each includes basic settings, pre-configured data, plug-ins, and plug-in parameters.  See “Creating Data Store Adapters” on page 669.

**Table 51** Contents of the Data Anywhere Toolkit for Integrating NetPoint and VDE

Subdirectory	File Components	Description
\mapping_templates	OblixADAMMapping_mpy.xml OblixADMapping_mpy.xml OblixDBMapping_mpy.xml OblixeDirectoryMapping_mpy.xml OblixSunOneMapping_mpy.xml	Mapping script templates. These sample mappings achieve the same configuration as those produced by the Object Class Mapper plug-in in the adapter template. Mapping scripts are more flexible and can produce a fine level of adjustment not available through the plug-in.  See "Creating Mapping Files for Adapters" on page 667.
\conversion_tools	obmigrateDN.exe obmigrateDNmsg.lst	The DN conversion binary and configuration file. When you integrate VDE with existing COREid installations, obmigrateDN converts the user and group DN in the NetPoint configuration tree by internally calling obmigratedata for handling the VDE DN specific operations, which then refers to the ldapmodify executable.
<b>Directory</b>  \migration_tools \obmigratedata	oc_DN_Conversion_map_osd_offline.lst at_DN_Conversion_map_osd_offline.lst oc_DN_Conversion_map_osd_online.lst at_DN_Conversion_map_osd_online.lst	Files required by obmigratedata during runtime. They have the objectclass and attribute details that need special handling. This is similar to typical upgrades, except the DN tool requires special handling in offline and online mode (as mentioned in the readme).
_install_dir\Oblix \apps\common\bin\	globalparams.xml	A file that controls the scope of searches in the searchbase, among other things.

**Table 51** Contents of the Data Anywhere Toolkit for Integrating NetPoint and VDE

Subdirectory	File Components	Description
<code>_install_dir\Oblix\lib</code>	obxmlengine.dll (windows) obxerces-c21.dll (windows) msvci70.dll (windows) msvci70d.dll (windows) msvcr70.dll (windows) msvcr70d.dll (windows) libxmlengine.so (solaris) libstdc++.so.5 (solaris) libgcc_s.so.1 (solaris) all required ldap sdk libraries for solaris	

## About the DN Conversion Tool

The conversion tool for use with *existing* NetPoint installations is provided in the DataAnywhere toolkit that you download when you retrieve the NetPoint 7.0.2 installation packages from the Oblix Web site:

`DNConversionTool_install_dir\oblix\tools\DataAnyWhere\conversion_tools\obmigrateDN.exe`

---

**Important:** The DNConversion Toolkit is a stand-alone package that you *must* download from the Oblix Web site to ensure that you have the latest version. Do *not* use the toolkit in the `COREid_install_dir`.

---

The conversion tool changes the native DNs in the configuration and policy branches of the NetPoint configuration and policy tree into logical (virtual) DNs that can be used by the virtual directory.

### Conditions

For the conversion to occur successfully, your NetPoint directory must meet the following two conditions:

- NetPoint configuration and policy data resides in a native directory outside VDE.
- NetPoint must see this configuration and policy data as belonging to a directory distinct from the VDE virtual directory, which contains all the user data that NetPoint sees.

## Requirements

- A file containing the list of DN attributes to be converted
- A mapping list you create, which correlates native DNs to logical DNs
- Host name, Port number, Bind DN, Password, Directory type, Config DN, Oblix node, Install Dir, Native DN, Logical DN, Mode (online, offline, test)

## Details

- The tool only performs conversion when the *domains* differ in VDE. For example:

If the DN on NetPoint is:

`o=company, c=us`

And the DN for iPlanet on VDE is:

`o=iPlanet, o=company, c=us`

Then the conversion takes place by referring to the mapping details given as input.

- If only the attributes themselves differ, no mapping occurs. For example:

`cn=manisha, o=company, c= us`

cannot be mapped to VDE

`cn=manisha, o=iPlanet, o=company, c=us`

- You must run the tool at least once to convert each DN value.

---

**Note:** If the configuration branch is *not* on the same directory server as the policy branch, you must run the tool twice for each DN value.

---

- The tool does not support SSL.
- Loading of the DSML version of the schema is not automated.
- If you are upgrading an existing NetPoint installation, before re-running NetPoint setup you must manually remove the DBProfile branch from the directory being integrated.

---

**Note:** Be sure to remove old DBProfiles manually before NetPoint setup; after setup the new DBProfiles are created automatically

---

- The tool does not support Active Directory Services Interface (ADSI); you must use SSL instead, if you want a secure connection.

# NetPoint-VDE Integration Templates

Oblix provides *adapter templates* and script templates to assist you with quick setup of each directory and a database. When you configure an adapter, you can choose a template described below to complete schema mapping and special handling. Depending on the mapping criteria, these templates can be used as they are or they can provide a base for tailoring.

The provided templates are listed in Table 51 on page 704. To fully understand what each template can achieve, see discussions below:

- “Templates for Active Directory” on page 708
- “Templates for ADAM” on page 711
- “Templates for Sun Directory Server” on page 714
- “Templates for eDirectory” on page 715
- “Database Template: OblixDBAdapterUsingScript” on page 716
- “Schema Mapping Script Templates” on page 716

---

**Note:** ObjectClass Mapper templates are a plug-in with parameter-based user interface. ObjectClass mapper templates and script templates perform the *same* operations and produce the same results. Using the script may be preferable to some and provide greater freedom while using the mapper may be preferable to others. The flexibility is yours to choose.

---

For additional information, see “Creating Data Store Adapters” on page 669 and “About the Data Anywhere Toolkit” on page 703.

## Templates for Active Directory

NetPoint provides three templates for use with Active Directory:

- “OblixADAdapterUsingMapper for Active Directory” on page 709
- “OblixADAdapterUsingScript for Active Directory” on page 711
- “OblixADSSLAdapterUsingMapper for Active Directory” on page 711

## OblixADAdapterUsingMapper for Active Directory

This template defines an adapter that converts Active Directory user and group data to VDE inetorgperson and groupofuniquenames using the Object Mapper plug-in, which includes the following:

1. A mechanism to set the DN attributes with all attributes that have DN syntax from inetorgperson, groupofuniquenames, and NetPoint user auxiliary classes to ensure these DNs are stored in the native DN format.
2. A plug-in for Active Directory ranged attributes  
Active Directory returns the entry as *xxx* bytes chunks. This plug-in concatenates all the chunks into a single result entry.
3. A plug-in for the ObjectClass Mapper, which provides a parameter-based user interface for object class and attribute mappings as described in Table 52, is also included.

**Table 52** OblixADAdapterUsingMapper, ObjectClass Mapper Plug-in Parameters

parameter	value	comment
filterObjectClassOnModify	true	Assume Active Directory is configured as static auxiliary class.
addAttribute-group	samaccountname=%cn%	If the object class is group, add attribute samaccountname and set the value to be equal to cn. This is because Active Directory requires samaccountname for group.
addAttribute-group	groupType=4	If the object class is group, add attribute groupType and set the default value to 4. The value can be changed based on customer needs.
mapAttribute	uniqueMember=member	Map VDE attribute uniqueMember to Active Directory attribute member.
mapAttribute	owner=managedBy	Map VDE attribute owner to Active Directory attribute managedBy.
mapAttribute	uid=samaccountname	Map VDE attribute uid to Active Directory attribute samaccountName.
filterAttribute-group	see Also, businessCategory,o	If the object class is group, filter out these attributes. This is because Active Directory group does not support these three attributes.
mapObjectClass	groupofuniquenames=group	Map VDE objectclass groupOfUniqueNames to Active Directory objectclass group.

**Table 52** OblixADAdapterUsingMapper, ObjectClass Mapper Plug-in Parameters

parameter	value	comment
mapObjectClass	inetorgperson=user	Map VDE objectclass inetorgperson to Active Directory objectclass user.
filterAttribute	(list of system attributes)	Filter out all the attributes in the list. Active Directory has a long list of system attributes that we don't want NetPoint to see.
directoryType	ActiveDirectory	The directory type.
activationAttribute	obuseraccountcontrol	The Oblix attribute name that the Active Directory adapter should use to find for activation and deactivation. The Active Directory adapter then sets the native flag useraccountcontrol based on this.
activationValue	ACTIVATED	The activation value of obuseraccountcontrol.
deactivationValue	DEACTIVATED ObWfPendingActivate ObWfPendingDeactivate	The deactivation values of obuseraccountcontrol.
filterAuxiliaryClass	person, organizationalPerson OblixOrgPerson, oblixpersonpwdpolicy oblixadvancedgroup oblixgroup, oblixAuxLocation	The auxiliary classes to be filtered out. This is based on the assumption that the Active Directory is configured as static auxiliary class.

4. A plug-in for the Active Directory password, which requires the use of the SSL connection to set or change the user password using the parameters in Table 53, is included.

**Note:** If the current adapter is using the open connection, this plug-in redirects password set/change to an adapter configured with an SSL connection.

**Table 53** Active Directory Password Plug-in Parameters and Values

parameter	value	comment
adapter	AD SSL Adapter	Redirect to the adapter defined in template OblixADSSLAdapterUsingMapper.
mapPassword	(not set. Default is true)	Map password attribute from userPassword to unicodePwd.

## **OblixADAdapterUsingScript for Active Directory**

This template (OblixADAdapterUsingScript) achieves *exactly* the same result as described above, and includes the same items described in 1, 2, and 4 of “OblixADAdapterUsingMapper for Active Directory” on page 709.

The only difference when using the OblixADAdapterUsingScript is item 3, which in this case will be:

3. A plug-in *script* written in Python, defined in the mapping script template OblixADMapping, with the parameters in Table 52 accomplishes everything stated for the ObjectClass Mapper in item 3 of “OblixADAdapterUsingMapper for Active Directory” on page 709.

## **OblixADSSLAdapterUsingMapper for Active Directory**

This template defines an adapter that connects to the Active Directory through SSL. This is for the *redirected* adapter identified in item 4 of discussions above. See:

- “OblixADAdapterUsingMapper for Active Directory” on page 709
- “OblixADAdapterUsingScript for Active Directory” on page 711

## **Templates for ADAM**

Three templates are provided for ADAM:

- “OblixADAMAdapterUsingMapper for ADAM” on page 711
- “OblixADAMAdapterUsingScript for ADAM” on page 713
- “OblixADAMSSLAdapterUsingMapper for ADAM” on page 713

## **OblixADAMAdapterUsingMapper for ADAM**

This template (OblixADAMAdapterUsingMapper) defines an adapter that converts ADAM user and group data to VDE inetorgperson and groupofuniquenames using the Object Mapper plug-in, which includes the following:

1. A mechanism to set the DN attributes with all attributes that have DN syntax from inetorgperson, groupofuniquenames and NetPoint user auxiliary classes to ensure these DNs are stored in the native DN format.

2. A plug-in for the Active Directory Ranged Attributes.  
ADAM also returns the entry as xxx bytes chunks. This plug-in concatenates all the chunks to a single-result entry.
3. A plug-in for the ObjectClass Mapper, which provides a parameter-based user interface for object class and attribute mappings, as described in Table 54 is also included.

**Table 54** OblixADAMAdapterUsingMapper, ObjectClass Mapper Parameters and Values

parameter	value	comment
filterObjectClassOnModify	(not set. Default is false)	Assume dynamic auxiliary class. Set it to true if ADAM is configured as static auxiliary class.
addAttribute-group	groupType=4	If the object class is group, add attribute groupType and set the default value to 4. The value can be changed based on customer needs.
mapAttribute	uniqueMember=member	Map VDE attribute uniqueMember to ADAM attribute member.
mapAttribute	owner=managedBy	Map VDE attribute owner to ADAM attribute managedBy.
mapAttribute	uid=samaccountname	Map VDE attribute uid to ADAM attribute samaccountName.
filterAttribute-group	seeAlso,businessCategory,o	If the object class is group, filter out these attributes. This is because Active Directory group does not support these three attributes.
mapObjectClass	groupofuniquenames=group	Map VDE objectclass groupOfUniqueNames to Active Directory objectclass group.
mapObjectClass	inetorgperson=user	Map VDE objectclass inetorgperson to ADAM objectclass user.
filterAttribute	(list of system attributes)	Filter out all the attributes in the list. ADAM has a long list of system attributes that we don't want NetPoint to see.
directoryType	ADAM	The directory type.
activationAttribute	obuseraccountcontrol	The Oblix attribute name that the ADAM adapter should use to find for activation and deactivation. The ADAM adapter then sets the native flag useraccountcontrol based on this.
activationValue	ACTIVATED	The activation value of obuseraccountcontrol.
deactivationValue	DEACTIVATED ObWfPendingActivate ObWfPendingDeactivate	The deactivation values of obuseraccountcontrol.

**Table 54** OblixADAMAdapterUsingMapper, ObjectClass Mapper Parameters and Values

parameter	value	comment
filterAuxiliaryClass	(not set)	The auxiliary classes to be filtered out. This is based on the assumption that the ADAM is configured as dynamic auxiliary class. Give auxiliary class names for this parameter if ADAM is configured as static auxiliary class.

- A plug-in for the Active Directory password (ADAM requires the use of the SSL connection to set or change the user password using the parameters in Table 55), is included.

---

**Note:** If the current adapter is using the open connection, this plug-in redirects password set/change to an adapter configured with an SSL connection.

---

**Table 55** OblixADAMAdapterUsingMapper, ActiveDirectory Password Parameters

parameter	value	comment
adapter	ADAM SSL Adapter	Redirect to the adapter defined in template OblixADAMSSLAdapterUsingMapper.
mapPassword	false	Do no map password attribute because ADAM uses attribute userPassword.

## OblixADAMAdapterUsingScript for ADAM

This template (OblixADAMAdapterUsingScript) achieves *exactly* the same result as described above, and includes the same items described in 1, 2, and 4 of “OblixADAMAdapterUsingMapper for ADAM” on page 711.

The only difference when using the OblixADAMAdapterUsingScript is item 3, which in this case will be:

- A plug-in *script* written in Python, defined in template OblixADAMMapping, with the parameters in Table 54 accomplishes everything stated for the ObjectClass Mapper in item 3 in “OblixADAMAdapterUsingMapper for ADAM” on page 711.

## OblixADAMSSLAdapterUsingMapper for ADAM

This template defines an adapter that connects to the ADAM directory through SSL. It is for the *redirected* adapter identified in item 4 of discussions above. See:

- “OblixADAMAdapterUsingMapper for ADAM” on page 711
- “OblixADAMAdapterUsingScript for ADAM” on page 713

## Templates for Sun Directory Server

Two templates are provided for the Sun Directory Server:

- “OblixSunOneAdapterUsingMapper for SunOne” on page 714
- “OblixSunOneAdapterUsingScript for SunOne” on page 714

### OblixSunOneAdapterUsingMapper for SunOne

This template defines an adapter that converts Sun Directory Server (formerly SunOne) inetorgperson and groupofuniquenames to VDE inetorgperson and groupofuniquenames using the Object Mapper plug-in, which includes the following:

1. A mechanism to set the DN attributes with all attributes that have DN syntax from inetorgperson, groupofuniquenames and NetPoint user auxiliary classes. This is to ensure these DNs are stored in the native DN format.
2. A plug-in (ObjectClass Mapper), which provides a parameter based user interface for object class and attribute mappings as shown in Table 56, is also included.

**Table 56** OblixSunOneAdapterUsingMapper, ObjectClass Mapper Parameters

Parameter	Value	Comment
directoryType	SunOne	The directory type.
activationAttribute	obuseraccountcontrol	The Oblix attribute name that the SunOne adapter should use to find for activation and deactivation. The SunOne adapter then sets the native flag nsaccountlock based on this.
activationValue	ACTIVATED	The activation value of obuseraccountcontrol.
deactivationValue	DEACTIVATED, ObWfPendingActivate, ObWfPendingDeactivate	The deactivation values of obuseraccountcontrol.

### OblixSunOneAdapterUsingScript for SunOne

This template (OblixSunOneAdapterUsingScript) achieves *exactly* the same result as described above, and includes the same item described in 1 of “OblixSunOneAdapterUsingMapper for SunOne” on page 714.

The only difference when using the OblixSunOneAdapterUsingScript is item 2, which in this case will be:

2. A *script* written in Python, defined in template OblixSunOneMapping, with the parameters in Table 56 accomplishes everything stated for the ObjectClass Mapper in item 2 of “OblixSunOneAdapterUsingMapper for SunOne” on page 714.

## Templates for eDirectory

Two templates are provided for eDirectory:

- “OblixeDirectoryAdapterUsingMapper for eDirectory” on page 715
- “OblixeDirectoryAdapterUsingScript for eDirectory” on page 716

### OblixeDirectoryAdapterUsingMapper for eDirectory

This template defines an adapter that converts eDirectory inetorgperson and groupofuniquenames to VDE inetorgperson and groupofuniquenames using the Object Mapper plug-in, which includes the following:

1. A mechanism to set the DN attributes with all attributes that have DN syntax from inetorgperson, groupofuniquenames and NetPoint user auxiliary classes. This is to ensure these DNs are stored in the native DN format.
2. A plug-in (ObjectClass Mapper), which provides a parameter based user interface for object class and attribute mappings as shown in Table 57.

**Table 57** OblixeDirectoryAdapterUsingMapper for eDirectory

parameter	value	comment
directoryType	SunOne	The directory type.
activationAttribute	obuseraccountcontrol	The Oblix attribute name that the eDirectory adapter should use to find for activation and deactivation. The eDirectory adapter then sets the native flag logindisabled.
activationValue	ACTIVATED	The activation value of obuseraccountcontrol.
deactivationValue	DEACTIVATED, ObWfPendingActivate, ObWfPendingDeactivate	The deactivation values of obuseraccountcontrol.

## **OblixeDirectoryAdapterUsingScript for eDirectory**

This template (OblixeDirectoryAdapterUsingScript) achieves *exactly* the same result as stated above using OblixeDirectoryAdapterUsingMapper.

The only difference when using the OblixeDirectoryAdapterUsingScript is item 2, which in this case will be:

2. A *script* written in Python, defined in template OblixeDirectoryMapping, with the parameters in Table 57 accomplishes everything stated for the ObjectClass Mapper in item 2 of “OblixeDirectoryAdapterUsingMapper for eDirectory” on page 715.

## **Database Template: OblixDBAdapterUsingScript**

This template defines an adapter for a database. It does *not* include specific mapping but *does* call the OblixDBMapping *script*. The script, defined in the template OblixDBMapping, is written in Python and filters out the unnecessary mention of objectclass during the LDAP operation.

## **Schema Mapping Script Templates**

The following *mapping script templates* are used by the adapter templates described above. These sample mappings achieve the same configuration as those produced by the Object Class Mapper plug-in in the adapter template. Mapping scripts are more flexible and can produce a fine level of adjustment not available through the plug-in

These mapping script templates provide a script alternative to accomplishing the schema mapping and special handling:

**OblixADMMapping**—This mapping template performs the following:

- Converts the Active Directory user and group to inetorgperson and groupofuniquenames, respectively
- Sets the native flag when a user is activated or deactivated
- Handles the static auxiliary objectclass
- Sets grouptype to 4
- Sets the useraccountname so that it is identical to cn.

**OblixADAMMapping**—This mapping template performs the following:

- Converts Active Directory user and group to inetorgperson/groupofuniquenames, respectively
- Sets the native flag when a user is activated or deactivated

- Handles the static auxiliary object class
- Sets grouptype to 4.

**OblixDirectoryMapping**—Sets the native flag when a user is activated or deactivated

**OblixSunOneMapping**—Sets the native flag when a user is activated or deactivated

## Integration Tips

The following section provides miscellaneous information to guide your NetPoint-VDE integration. See also “Database Connectivity Tips” on page 719.

**Mapping DN**—The mapped DN is the logical DN in VDE. However, there is no physical node for the mapped DN.

If the application (COREid for example) needs to search the logical DN to detect that DNs existence or to retrieve its attributes, the entry needs to be added manually (using the `ldp.exe` utility, for example). For example, if the mapping DN is `o=virtual company`, the corresponding entry needs to be created through `ldp.exe` so that:

- `objectclass: organization`
- `o: virtual_company`

where `organization` is `xxx`, and `virtual_company` is `xxx`.

**Reference DN in Configuration and Policy Data**—The reference DN such as a UID used in policy data, is in its logical form. This means that it is stored as the DN of VDE, not the native directory. As a result, once the VDE namespace mapping is completed, that mapping should *not* be changed. Changing the namespace mapping will impact the reference DNs stored in the NetPoint configuration and policy data.

**Schema Mapping**—When mapping an attribute from logical to native, be sensitive to the syntax and whether it is multi-valued or single-valued.

- For VDE to directory mapping, the syntax should be kept the same except for minor adjustments of string syntaxes.
- For VDE to database mapping, use Table 58 as a guideline:

**Table 58** VDE to Database Mapping

LDAP Attribute Syntax	MS SQL Data Type
1.3.6.1.4.1.1466.115.121.1.5 DESC 'Binary'	binary
1.3.6.1.4.1.1466.115.121.1.6 DESC 'Bit String'	binary

**Table 58** VDE to Database Mapping

1.3.6.1.4.1.1466.115.121.1.7 DESC 'Boolean'	varchar
1.3.6.1.4.1.1466.115.121.1.8 DESC 'Certificate'	binary
1.3.6.1.4.1.1466.115.121.1.9 DESC 'Certificate List'	binary
1.3.6.1.4.1.1466.115.121.1.10 DESC 'Certificate Pair'	binary
1.3.6.1.4.1.1466.115.121.1.11 DESC 'Country String'	varchar
1.3.6.1.4.1.1466.115.121.1.12 DESC 'DN'	varchar
1.3.6.1.4.1.1466.115.121.1.15 DESC 'Directory String'	varchar
1.3.6.1.4.1.1466.115.121.1.22 DESC 'Facsimile Telephone Number'	varchar
1.3.6.1.4.1.1466.115.121.1.23 DESC 'Fax'	varchar
1.3.6.1.4.1.1466.115.121.1.24 DESC 'Generalized Time'	timestamp
1.3.6.1.4.1.1466.115.121.1.26 DESC 'IA5 String'	binary
1.3.6.1.4.1.1466.115.121.1.27 DESC 'INTEGER'	Numeric / int
1.3.6.1.4.1.1466.115.121.1.28 DESC 'JPEG'	binary
1.3.6.1.4.1.1466.115.121.1.33 DESC 'MHS OR Address'	varchar
1.3.6.1.4.1.1466.115.121.1.36 DESC 'Numeric String'	varchar
1.3.6.1.4.1.1466.115.121.1.38 DESC 'OID'	varchar
1.3.6.1.4.1.1466.115.121.1.39 DESC 'Other Mailbox'	varchar
1.3.6.1.4.1.1466.115.121.1.41 DESC 'Postal Address'	varchar
1.3.6.1.4.1.1466.115.121.1.44 DESC 'Printable String'	varchar
1.3.6.1.4.1.1466.115.121.1.50 DESC 'Telephone Number'	varchar
1.3.6.1.4.1.1466.115.121.1.53 DESC 'UTC Time'	timestamp

## Database Connectivity Tips

Following are several OctetString database connectivity considerations:

- Entry Name Formation
- Multi-Table Writes
- Multi-Value Attributes
- Searches
- Writes
- Cascading Deletes

**Entry Name Formation**—All database fields that will be used as part of an entry's name (with the exception of the “base” part of the entry's name) must be contained in the rows that are mapped and returned to VDE via the database adapter.

For example, if a hierarchy is being created in which user objects will contain both a common name (cn) and an organizational unit (ou) in their name (cn=Joe User,ou=Marketing), both cn and ou must be part of the entry being created.

In pure LDAP, the ou attribute would not be required as it is part of the parent entry. Since databases are not hierarchical, this is the only reasonable way to support this functionality without requiring considerable new metadata be created and managed to define hierarchy.

**Multi-Table Writes**—Multi-table writes are not possible directly to a single database adapter. This is not a design limitation of VDE, but rather a practical database limitation. For example, views in most databases cannot be updated directly when they present multiple tables.

VDE gets around this limitation through the use of its own Join View implementation. By creating multiple database adapters (perhaps one for each table) and defining the relationship between them, it is possible to have VDE manage writes to entries that are constructed through multiple tables. Further information on creating Join Views can be found in OctetString's product manual.

**Multi-Value Attributes**—Databases typically do not allow for multiple values for a single field within a single table row. There are exceptions where an array type is supported, but these data types tend to be relatively limited. Some users put multiple values into a row by separating data (such as account flags) within a field using delimiters such as commas or pipes (|).

Traditional database design states that fields that would have multiple values should be normalized into an additional table. Databases that are part of a data warehouse may take a different approach in which every permutation of every field is placed into a denormalized table.

VDE can generally support either model. NetPoint suggests using the “permutations in a denormalized table” method *only* as an absolute last resort. The normalized, secondary table approach probably won’t return consistently accurate results for NetPoint searches.

For more information about multi-valued attributes, see:

- “NetPoint-VDE Integration Templates” on page 708
- “Multi-Value Attribute Problems” on page 721

**Searches**—Searches are supported to normalized or denormalized tables without doing anything other than configuring a database-level join as necessary.

The one consideration that should be kept in mind on searches is that due to the most popular use cases, the current design of the database adapter is that if a multi-value attribute is searched to only return the value that matched the search as part of the entry. This facilitates high performance large group searches. OctetString expects to make this configurable (to support doing a subselect to return all values) in a future version. An additional search can be performed in mapping to retrieve all attributes in the mean time.

**Writes**—Writes to normalized tables must be performed via a Join View that splits out attributes to each table based on the design of the database. This is required since while there are general guidelines for database design, every customer's database is different.

Most OctetString customers that use existing and important tables also use stored procedures as part of controlling updates to those tables. These are akin to API calls and are proprietary to each database in the way they are constructed and called, while the calls themselves are proprietary to the customer. OctetString supports stored procedures through the use of its plug-in system.

OctetString can manage direct writes to denormalized tables that have each value for the field that will be used in the entry associated with the field used as the RDN for the entry. Add, Modify, and Delete are all supported.

Within the modify operation, it should be noted that the way the modify->replace works is to remove existing attribute values and add new ones. This translates into a SQL delete and a SQL insert rather than a complex set of SQL inserts, updates, and deletes. The potential issue here is with customers that have normalized tables in which the insert and/or delete will trigger other actions within the database. In such a plugin would need to be constructed that would handle the modify->replace operation.

Most OctetString customers do not run into this as they either are not using direct SQL access for changes, are using multiple values only for read, or are only using modify->add and modify->remove directly. For example, customers solving issues

with big groups are storing groups in databases via VDE. Most group membership changes are adds and removes rather than replaces.

**Cascading Deletes**—Of the issues mentioned in writes, the biggest thing to watch for in an existing database where VDE will be handling database writes directly is a customer database’s use of cascading deletes. With cascading deletes it is possible that a modify->replace as documented in the previous section would trigger deletes outside of the table being directly affected.

This said, if the trigger is based on the normalized table, this is not an issue as when modifying the single-valued normalized table VDE will do an SQL update rather than a delete-insert sequence. This should eliminate the issue mentioned.

If in doubt about what will happen with a particular customer situation with a potentially dangerous setup involving Cascading Deletes, contact OctetString. You can also turn up the debug level on VDE and point it to a test database to see the SQL being generated by VDE for any sequence of LDAP operations.

## Troubleshooting

You should be aware of several conditions that might affect your NetPoint-VDE integration environment:

- “Directory Server Problems” on page 721
- “Multi-Value Attribute Problems” on page 721
- “Secondary Data Store Problems” on page 722
- “Unexpected Group Deletion Problem” on page 723

### Directory Server Problems

**Active Directory or ADAM Search Problem**—With VDE and Active Directory or ADAM directory servers, you cannot search with the “That Sounds Like” operator.

**Cause**—Active Directory or ADAM directory servers do not support the “That Sounds Like” search.

**Workaround**—Do not use the “That Sounds Like” search with Active Directory or ADAM directory servers.

### Multi-Value Attribute Problems

You cannot modify multi-value attributes through a Change Attribute workflow.

**Cause**—The default VDE schema includes multi-valued attributes, as does the Sun Directory Server schema. The attribute syntax on Active Directory sometimes may

not match. For example, the mail address in Active Directory is single-valued but on a Sun Directory Server and VDE this is multi-valued.

For example, when VDE is communicating with Active Directory and a Sun directory server if you create a Change Attribute workflow and try to change a multi-valued attribute (such as an mail address) for a user on the Sun Directory Server, the attribute is changed but on Active Directory the commit step fails and the attribute does not get changed.

**Workaround**—Do *not* change multi-valued attributes.

## Secondary Data Store Problems

1. **Sub-tree Search**—With a database split profile, you cannot derive an attribute from an attribute that is present in the secondary table.

**Cause**—NetPoint cannot perform a sub-tree search on an attribute from a secondary data store.

Suppose, for example, if you used the mapping template CustomOracleDBMapping\_mpy.xml and defined a derived attribute for genSiteOrgPerson as follows:

- Attribute Name: MyAttr
- Display Name: MyAttr
- Match Attribute: employeenumber
- Lookup Attribute: employeenumber
- Object class: genSiteOrgPerson

When you search for a user (Rohit for example) and view his profile, you can see the value for the employeenumber attribute but the myAttr value is blank.

In the example below, there is a database and a split profile and the following adapter templates:

```
CustomOracleAdaptorSplitPrimary_adapter_template.xml
CustomOracleAdaptorSplitSecondary1-1_adapter_template.xml
CustomOracleAdaptorSplitSecondary1-M_adapter_template.xml
CustomAdapterJoinView_adapter_template.xml
```

**Workaround**—Do *not* configure attributes from a secondary data store.

2. **Create User Workflow**—When defining a Create User workflow, NetPoint allows you to select attributes from the VDE secondary view. At run time, the user entries are created in the primary view; however, the workflow fails and these entries cannot be used by NetPoint.

**Cause**—NetPoint gets all the attributes from VDE and therefore has no knowledge about which attributes are obtained from the primary data store, rather than the secondary data store.

**Workaround**—Do *not* configure attributes from a secondary data store.

## Unexpected Group Deletion Problem

When you set up NetPoint to integrate with a VDE virtual directory that federates at least one LDAP directory and at least one *database* table, then you try to remove a member from a group in the LDAP directory, the entire group is removed from that directory.

**Cause**—For performance reasons, VDE returns to NetPoint only the member you specify for deletion. By contrast, a “standard” LDAP directory server would return all the members in the group.

This non-standard VDE behavior has consequences when you try to use NetPoint to delete a member from a group. Because a “standard” LDAP directory server returns all members of the group, NetPoint stills “sees” the rest of the members of the group after the one member has been deleted. But since VDE has returned to NetPoint only the single member of the group designated for deletion, NetPoint does not “see” any other group members after it deletes the returned member, so it assumes that the group is now empty and it deletes the group and all its members.

---

**Important:** This is generic to *all* DN attributes, *not* just uniqueMember of a group. The workaround must be applied to all DN attributes where multiple values are a possibility.

---

**Work Around**—See the customized file shown in Figure 43, "Sample Mapping Customized for an Oracle Database" on page 688 and pay attention to the details shown below to prevent COREid from writing dummy user to backend database:

---

workaround to prevent COREid from writing dummy user to backend database

```
if haveAttributeValue('uniqueMember','cn=Dummy User'):
    #removeAttributeValue('uniqueMember','cn=Dummy User')
    if operation != 'modify':
        removeAttributeValue('uniqueMember','cn=Dummy User')
    else:
        change = removeAttribute('uniqueMember')[0]
        change.values.remove(DistinguishedName('cn=Dummy User'))
        addEntryChange(change)
```

---



# Index

## Symbols

.NET Passport  
DotNetGate, installing 502

## A

AAT *See* Application Assembly Tool  
About NetPoint and the AzMan Plug-In 507  
About NetPoint and the SharePoint Portal Server 585  
About Windows Impersonation 586  
Access Manager 500  
    ACLs, management of 33  
    definition 195  
    WebLogic groups and roles, management role 33  
Access Manager API  
    and WebLogic resources 28  
Access Server 500  
Access Server API  
    and WebLogic resources 29  
    and WebSphere resources 183  
Access Server SDK 183  
Access Server, installation and configuration  
    about configuring 195  
    clock synchronization, importance of 195  
Access System  
    NetPoint BEA Ready Realm automatic configuration  
        37  
Access System Console  
    definition 195  
AccessGate  
    *See also* NetPoint BEA Ready Realm  
    definition 195  
    for NetPoint BEA Ready Realm configuration 35  
AccessGate and Access Server  
    about configuring 195  
AccessGate, installation and configuration  
    clock synchronization, importance of 195  
    NetPoint BEA Ready Realm configuration 51  
ACLs  
    *See also* protected resources  
    about 32  
    Access Manager 33  
    EJB or WebApp, protecting 70-73  
    initial generation 32  
    policies generated for 38  
    resource protection sample 87-88  
    troubleshooting 93  
ActivCard 485  
Active Directory and NetPoint BEA Ready Realm  
    automatic policy creation 88  
    create\_user workflow, problem 89

    NetPointBEARealm.properties file, domain setting 89  
Active Directory and NetPoint Connector for WebSphere  
    NetPointWASRegistry.properties file, domain setting  
        295  
Active Directory Forest  
    SecurID considerations 472  
Active Directory forest  
    and NetPoint BEA Ready Realm 88  
anonymous users, enabling 385  
ASP.NET 539  
attribute flow 567  
attributes, workflows  
    NetPoint BEA Ready Realm workflows 79-81  
audience 19  
authentication  
    Access Server API and WebLogic resources 29  
    Access Server API and WebSphere resources 183  
authentication schemes  
    WebLogic resources 37  
Authentication with Smart Card 490  
AuthenticationMode 242  
authorization  
    Access Server API and WebLogic resources 29  
    Access Server API and WebSphere resources 183  
Authorization Manager 516  
Authorization Stores 516  
Authorization with the NetPoint AzMan Plug-In 509  
Authorization with the NetPoint Security Connector for  
    ASP.NET 543  
AzMan Plug-in Parameters 514

## B

BEA  
    integration overview 28  
BEA Ready Realm  
    Objects, mapping 31  
    supported platforms 352  
    supported versions 352  
BEA WebLogic Portal. *See* WebLogic Portal  
BEA WebLogic Server. *See* WebLogic Server  
Bind DN 664  
Broker role, sample descriptor 83

## C

cache, NetPoint BEA Ready Realm  
    group caching and user caching 50  
    size specification 47  
caching realm  
    creating for WebLogic custom realm 56  
    security file realm, creating 58  
cert\_authn.dll 490  
cert\_decode with Smart Card authentication 492  
CLASSPATH  
    "All jars are not in classpath" error 90

- modifying for WebLogic Portal Servers 64-65
- modifying for WebLogic resources 45
- modifying for WebLogic Servers 63-64
- NoClassDefFound error 91
- Client Certificate Authentication Scheme for Smart Card 491
- CMR 183, 235, 244
- com.oblix.access.ObAccessException, error message 91
- com.oblix.accessmgr.ObAMException, error message 91
- com.oblix.portalSSO.OblixLoginHelper class 76
- com.oblix.realm.NetPointBEAR realm, error message 91
- configuration files
  - banner.asp 361
  - ConfigServices.properties 246
  - dologout.asp 361
  - extautherror.html, example of 431
  - global.srvc, example of 428
  - Login.html, example of 429
  - NetPointWASRegistry.properties 283
  - Oblix.srvc, example of 428
  - redirect.html, example of 431
  - trustedservers.properties 292
  - webgate.properties 290
  - WGate.config, example of 427
- configure the authentication scheme for Smart Card 487
- configuring single sign-on 359
  - config.xml 359
  - oblix.asp 359
- Configuring the Access System for WAS Integration 195
- contact information 21
- COREid Server and WebPass
  - instance, role of 192
- Creating an Authorization Scheme for the AzMan Plug-In 532
- credential\_mapping with Smart Card authentication 492
- Custom Member Repository See CMR 235

## D

- Data AnyWhere 621
- data synchronization 363
- deployment descriptors, role mapping sample 85-86
- descriptors, samples of 83, 85-86
- DotNetGate 502
- Dynamic Groups, importing 393
- DynamicRoleMapServlet 86

## E

- EJB
  - mapping EJB roles to NetPoint roles 85-86
  - protecting ??-73
- ejb-jar.xml 66

## F

- fileRealm.properties file, troubleshooting 93

## G

- gadgets
  - adding to guest pages 384
  - creating 379
  - displaying 383
  - exporting 380
  - gateway, using 392
  - importing 380
  - pop-up window 392
  - using 378
- getAttributes 184, 235
- getGroupMemberships 184, 235
- getUser.minimum.attributes 237, 243
- Global Catalog
  - AD forest
    - create\_user workflow, troubleshooting 89
- group classes, WebLogic to NetPoint mapping 32
- group.baseattributes 237, 243
- GroupSrvCenter 290
- guest access, enabling 385
- guest user, specifying login ID 50

## H

- Host name 664

## I

- Identity XML 183
  - WebSphere Portal administration, role in 183
- IdentityXML
  - Identity XML-WebGate single sign-on, enabling 45
  - NetPoint BEA RR, role of 44
  - WebLogic Portal administration, role in 29
- IIS Manager with Smart Card authentication 490
- Integrating NetPoint with Authorization Manager Services 507
- Integrating NetPoint with Smart Card Authentication 481
- Integrating SecurID Authentication 448
- Integrating the NetPoint Security Connector for ASP.NET 539
- InvalidCredential error 92
- IPrincipal.IsInRole Method Syntax 540

## J

- J2EE specification, BEA WebLogic Server 29
- J2EE\_Acl
  - EJB or WebApp, protecting 70-73

- policies generated for 38
- WebLogic to NetPoint mapping 32
- J2EE\_Role
  - created automatically 37
  - EJB or WebApp, protecting 66-70
  - location of 33
  - policy domain for, specifying 48
  - resource mapping 32
  - user caching 50
- java.security.AccessControlException 93
- JNDI name problem, troubleshooting 96
- join and projection rules 566
- Join rules
  - configuring 566
- jssecacert keystore 46, 218

## K

- keytool 46, 218

## L

- LDAP authentication source, creating 354
- login
  - as different user 387
  - into Netpoint 387
  - into Plumtree 387
- Login Framework, configuring for SSO 74, 78, 79
- LOGIN\_HELPER\_CLASS, adding to web.xml file 76

## M

- Member Services
  - CMR 183
- MemberSubSystem section 242
- MIIS database columns
  - configuring 564

## N

- Nested groups 289
- NetPoint
  - documentation 20
  - integrations 23
  - SecurID Access Server 440
- NetPoint BEA Ready Realm
  - ACL resource protection example 87-88
  - architecture 30
  - classes, mapping of 31
  - error messages and resolutions 90-98, 170-??
  - group caching 50
  - guest user, specifying login ID 50
  - security file realm, creating 58
  - single sign-on examples 81-82, 84-85
  - single sign-on, about 33

- troubleshooting 90-98, 170-??
- user caching 49
- workflow attributes 79-81
- workflows, generating automatically 39
- NetPoint BEA Ready Realm, configuration
  - automatic configuration 37
  - configuration information, generating 39
  - prerequisites 35
  - process overview 34, 52
- NetPoint BEA Ready Realm, installation
  - cache size specification 47
  - custom realm, creating as ??-58
  - J2EE policy domain, specifying 48
  - log file setup 47
  - setup tasks ??-52
  - WebPass and Access Server components 44-113, ??-113
- NetPoint BEA Ready Realm, integration
  - Access Manager API, role of 28
  - AccessGate configuration information 51
  - AccessGate, adding 35
  - certificate setup for WebPass SSL 45-46
  - incompatibility with Active Directory and automatic policy creation 88
  - integration use case 32
  - WebLogic Portal, integration with 30
  - WebLogic Server integration characteristics 29
  - WebLogic Server, integration with 29
- NetPoint CMR See CMR 235
- NetPoint Connector for WebSphere
  - architecture 186
  - configuration, prerequisites 190
  - error messages and resolutions 295
  - IdentityXML, role of 219
  - installation, setup tasks 208
  - integration
    - certificate setup for SSL 218
  - NetPoint components, installation of 206-215
  - supported platforms 352
  - transport security mode, specifying 212
  - troubleshooting 295
  - WebPass and Access Server components 219
- NetPoint Custom Member Repository See CMR 183
- NetPoint Dynamic Groups, importing 393
- NetPoint Role-Based Authorization Flow 549
- NetPoint single sign-on, testing 234
- NetPoint, installing 367
- NetPoint, setting up 367
- NetPointBEARealm.properties file
  - Activity Directory, domain setting 89
  - generating configuration entries 39
- NetPointCMR.jar 241
- NetPointResourceconf.conf 132
- NetPointWASRegistry 183
- NetPointWASRegistry.properties 283, 295
  - Active Directory, setting domain 295
- NoClassDefFound error 91

## O

- OB\_DynamicGroupsEnabled 290
- OB\_NestedGroupsEnabled 289
- obConfig.NO\_CONFIG\_FILE error message 90
- objects
  - WebLogic to NetPoint mapping 31
- Oblix MA
  - installation 558
- OblixAnonymous account, locking 387
- oblixgadgets.zip 380
- OblixHttpModule 543
- OblixPrincipal Object 543
- OBPrincipalHTTPModule 542
- ObSSOCookie
  - single sign-on sample 84-85
  - troubleshooting 95
  - WebLogic Portal Server configuration 74
  - WebSphere Portal, configuring for 245
- OctetString 621
- Oracle
  - architecture and integration 315
  - integration, overview 314
  - integration, testing 322
  - logout, specifying 321
  - Obsso.pkb 320, 333
  - troubleshooting 341, 344
  - versions supported 316
  - WebGate, installing 341, 344
- Oracle contact information 21
- Oracle SSO Server 314

## P

- Passport
  - DotNetGate, installing 502
- Password
  - Bind DN 664
- password management 396
- password.lst file 115
- personalization 184, 235
- personalizing user pages 387
- Plugin section 242
- pluginImplClass 242
- Plug-Ins for Smart Card Authentication 492
- Policy Domain for Smart Card Authentication 490
- policy domain, creating 368
- policy domains, J2EE roles
  - about 33
  - J2EE\_ACLS, generated for 38
  - specifying 48
- policy domains, WebLogic
  - domain, about 33
  - enabling 42
  - policies, regenerating 38
  - resources, creating automatically 37
- Port number 664

- post-installation tasks 218
- Preparing NetPoint for Smart Card Authentication 487
- Procedure
  - (MIIS) To configure extensions 569
  - (MIIS) To configure the attribute flow 567
  - (MIIS) To prepare for password synchronization 557
  - (MIIS) To set up a Run Profile for Delta Synchronization 570
  - (MIIS) To set up a Run Profile for Full Import 570
  - To add an impersonation action to your policy domain 599
  - To add the impersonation dll to your IIS configuration 600
  - To bind your trusted user to your WebGate 598
  - To configure IIS Security for the SPPS integration 604
  - To configure join and projection rules 566
  - To configure SSO for the WebSphere Portal v5 275
  - To configure SSO logout (WAS Portal v5) 275
  - To configure the MIIS database columns 564
  - To configure the SQL server connection 564
  - To configure the SQL Server Connection for the Oblix MA for MIIS 564
  - To configure the wildcard extension for SPPS virtual servers 605
  - To copy MIIS Files to COREid 571
  - To create a trusted user account 596
  - To create the Management Agent in MIIS 563
  - To define managed paths in SharePoint 594
  - To edit web.config for the SPPS integration 607
  - To give appropriate rights to the trusted user 597
  - To install NetPoint Components for SPPS integration 593
  - To install the Oblix Management Agent for MIIS 558
  - To integrate the WebSphere Portal v5 with NetPoint 260
  - To prepare for Oblix Management Agent for MIIS installation 557
  - To test impersonation through a Web page that displays server variables 603
  - To test impersonation through the Event Viewer 602
  - To test SSO for your NetPoint/SPPS integration 609, 619
  - To test your NetPoint/SPPS integration 608, 618
  - To test your OblixMA configuration 576
- Process overview
  - Access Server API operation with the NetPoint AzMan Plug-in 511
  - Authorization with the NetPoint CMR 238, 259
  - Authorization with the NetPoint Security Connector for ASP.NET 544
  - During Smart Card authentication 484
  - Events during authentication and authorization 549
  - Password synchronization (MIIS) 555
  - Provisioning (MIIS) 554
  - Request processing with SPPS integration 589
  - WebGate operation with the NetPoint AzMan Plug-in 509
- ProfileDataStorage 242

- Projection rules
  - configuring 566
- projection rules 566
- protected resources
  - EJB ??-73
  - WebLogic resources, protecting ??-73

## R

- related documentation 20
- Request Processing by the SPPS Integration 588
- resource types 117
  - ACLs 33
  - WebLogic definitions, creating automatically 37
- resources, protecting
  - EJB ??-73
  - WebLogic resources ??-73
  - WebLogic resources, about protecting ??-73
- RMI object, example of restricting access 87-88
- roles, EJB to NetPoint mapping sample 85-86
- RSA
  - ACE/Agent 434
    - requirements 440
  - ACE/Server 434
    - requirements 438

## S

- SampleGadgets.zip 379
- SAP
  - components 400, 414
  - extautherror.html, example of 431
  - global.srv, example of 428
  - integration, testing 421
  - Login.html, example of 429
  - NetPoint SSO, setting up 404, 418
  - Oblix.srv, example of 428
  - redirect.html, example of 431
  - supported versions 417
  - WGate.config, example of 427
- SAP ITS 400, 414
- SAP PAS 415
- SAP Portal
  - supported versions 417
- SAP portal
  - integrating with 422
- search users by attribute 184, 235
- searchbases
  - web\_logic\_system user, setting for 42
- SecurID
  - Active Directory Forest considerations 472
  - Add ACE/Server users to NetPoint 468
  - authentication 433, 434
    - forms 442
    - Plug-in 436
    - scheme Plug-ins 461

- challenge
  - method 459
  - parameters 460
- Create a SecurID authentication scheme 458
- Define an Authentication scheme for SecurID 461
- incorrect Passcode 439
- integration summary 437
- New PIN
  - mode 439
  - sequence 446
- Next Tokencode
  - mode 439
  - sequence 446
- platforms and requirements 437
- Plug-in parameters 469
- prerequisites 449
- Protect SecurID resources 464
- Set Up an ACE/Agent 450
- Set Up SecurID WebGate(s) 454
- tokencode 434
- Troubleshooting 476
- WebGate
  - requirements 441
- security file realm, creating 58
- Security Principals and Security Identifiers (SIDs) 540
- self-registration, enabling 396
- Setting Up Smart Card Authentication 485
- Setting Up the NetPoint Role Action for ASP.NET 548
- single sign-on 350
  - configuration files, editing 359
  - enabling 352
- single sign-on cookies
  - obSSOCookie, sample 84-85
  - obSSOCookie, troubleshooting 95
- single sign-on, integration
  - IdentityXML-WebPass SSO, enabling 45
  - WebGate users, LoginHelper class 76
  - WebLogic deployment descriptor, example 86-87
  - WebLogic Portal Server configuration 74
  - WebLogic Portal Server, configuring for 74, 78, 79
  - WebLogic resources 33
  - WebSphere Portal Server, configuring for 245
- single sign-on, samples
  - NetPoint BEA Ready Realm example 81-82
  - servlet sample 84-85
  - WebLogic and NetPoint BEA Ready Realm
    - deployment descriptor 86-87
- Smart Card Authentication with NetPoint 483
- Smart Card certificate enrollment 486
- Smart Card Challenge Method, Parameter, SSL 492
- Split profile 621
- SQL MA
  - creating in MIIS 563
- SQL Server Connection
  - Configuring for MIIS 564
- SSL
  - WebPass and NetPointBEA Ready Realm, certificate setup 45-46

- SSLPeerUnverifiedException error message 90
- sso authentication source, creating 352
- synchronizing data 363
  - automatic 363
  - manual 365
  - status, viewing 366

## T

- TAI *See* Web Trust Association Interceptor
- Task overview
  - (MIIS) Configuring the database, COREid, and SSO 571
  - Configuring the Management Agent 562
  - Configuring the NetPoint AzMan Plug-in 531
  - Installing the NetPoint Connector 206
  - Integrating NetPoint and the Portal Server 239
  - Integrating NetPoint with MIIS provisioning 555
  - Integrating NetPoint with SPPS includes 589
  - Integrating NetPoint with WAS v5 247
  - Integrating with Plumtree 351
  - Preparing for Oblix Management Agent installation 556
  - Preparing to configure NetPoint to use the AzMan Plug-in 532
  - Preparing to install the NetPoint Connector for WebSphere 190
  - Setting up NetPoint/SPPS Integration 604
  - Setting up Smart Card Authentication 485
  - Setting up the NetPoint Security Connector for ASP.NET 544
  - Using a COREid workflow to delete identities from the metaverse 583
- transport security integration
  - NetPoint BEA Ready Realm, specifying 51
  - NetPoint Connector for WebSphere, specifying for 212
  - WebLogic Custom Ready Realm, specifying 51
- transport security mode
  - NetPoint Connector for WebSphere, specifying for 212
- Troubleshooting
  - SecurID integration 476
- troubleshooting
  - ACLs and fileRealm.properties file 93
  - "All jars are not in classpath" error 90
  - fileRealm.properties file 93
  - InvalidCredential error 92
  - NetPoint BEA Ready Realm, error messages and resolutions 90-98, 170-??
  - NetPoint Connector for WebSphere, error messages and resolutions 295
  - NoClassDefFound error 91
  - obSSOCookie setting 95
  - single slash error message 90
  - WebLogic Portal Server, stockportal example 93
- trustedservers.properties 292
- two-factor authentication 434
- typographical conventions 21

## U

- um.properties 236
- um.properties file 242
- UnsatisfiedLinkError 91
- User classes, WebLogic to NetPoint mapping 32
- user pages, personalizing 387
- user.baseattributes 237, 242
- users
  - sample descriptor mapping 83
  - weblogic\_system user, changing 74
- Using the NetPoint AzMan Plug-In with the Access Server API 536
- Using the Security Connector for ASP.NET 544

## V

- VDE 621
  - Target data store 621

## W

- WAS *See* WebSphere Application Server
- Web Trust Association Interceptor
  - about 182
  - installing and configuring 226, 251
  - logging, enabling 229, 255
- web.xml file, adding LOGIN\_HELPER\_CLASS 76
- WebApp, protecting ??-73
- WebGate
  - single sign-on sample 84-85
- WebGate integration
  - NetPoint BEA Ready Realm single sign-on 33
- webgate.properties 290
  - location 227, 252, 283, 290
- WebLogic
  - applications not booting, troubleshooting 92
  - deployment descriptor example 86-87
  - JNDI name problem, troubleshooting 96
  - policies, regenerating 38
  - policy domain, enabling 42
  - resources, about protecting ??-73
  - roles, management of 33
  - single sign-on sample, running 81-82
- WebLogic Custom Security Realm
  - Access Manager API, role of 28
  - AccessGate configuration information 51
  - AccessGate, adding 35
  - architecture 30
  - cache size specification 47
  - classes, mapping of 31
  - group and user cache 50
  - guest user, specifying login ID 50
  - integration use case 32
  - J2EE policy domain, specifying 48
  - log file setup 47

- NetPointBEARealm.properties file 39
- single sign-on 33
- workflows, generating automatically 39
- WebLogic Custom Security Realm, configuration
  - automatic configuration 37
  - configuration information, generating 39
  - prerequisites 35
  - process overview 34
  - WebLogic configuration, process overview 52
- WebLogic Custom Security Realm, installation
  - NetPoint components, installation of 44-113, ??-113
  - setup tasks ??-52
- WebLogic Portal
  - about 30
  - groups and ACLs, initial generation 32
  - Identity XML as administration tool 29
  - InvalidCredential error 92
  - NetPoint BEA Ready Realm integration 29, 30
  - policy domain, creating automatically 37
  - resource type definitions, creating automatically 37
  - single sign-on 33
- WebLogic Portal Server
  - blank page, stockportal example, troubleshooting 93
  - CLASSPATH statement modifications 45
  - single sign-on, configuring for 74, 78, 79
  - startup script, modifying 64-65
- WebLogic Server
  - about 29
  - custom realm, creating ??-58
  - groups and ACLs, initial generation 32
  - InvalidCredential error 92
  - NetPoint BEA Ready Realm integration characteristics 29
  - policy domain, creating automatically 37
  - resource type definitions, creating automatically 37
  - server resources 29
  - startup script, modifying 63-64
- weblogic.log file 47
- weblogic\_system user
  - about changing 47
  - Active Directory and AD forest, incompatibility with 88
  - changing 74
  - creation of 37
- weblogic-ejb-jar.xml 66
- WebPass
  - about 192
  - COREid Server, role of 192
  - IdentityXML-WebPass single sign-on, enabling 45
  - installing for NetPoint BEA Ready Realm 44-113, ??-113
  - NetPoint Connector for WebSphere, installing for 219
- WebSphere
  - secure application, building 232
  - SimpleSessionSecure application, configuring 232
- WebSphere Application Server
  - about 182
  - security role mapping, testing 226
  - security roles, mapping NetPoint users and groups 188
  - supported platforms 189, 316
- WebSphere integration
  - configuration files 283, 426
- WebSphere Portal
  - about 235
  - access control, for 245
  - password management 244
  - single sign-on logout, configuring 245
  - single sign-on, configuring 245
  - supported platforms 189, 316
  - users and groups, managing 244
- WebSphere Portal Server
  - CMR 183
  - single sign-on, configuring for 245
- wms.xml 236
- wms.xml file 242
- workflows
  - WebLogic workflows, generating automatically 39

